AD-A073 901 NAVAL SURFACE WEAPONS CENTER WHITE OAK LAB SILVER SP--ETC F/G 16/2
MODIFLY: A MODULAR MULTI-DEGREE-OF-FREEDOM TRAJECTORY PROGRAM.(U)
MAR 79 J E HOLMES
UNCLASSIFIED NSWC/WOL/TR 78-58

1 OF 2
AD
A073901

(14) NSWC/WOL/TR 78-59

(11) (9) LEVEL II

AD A073901

(6)

MODIFLY: A MODULAR MULTI-DEGREE-OF-FREEDOM TRAJECTORY PROGRAM.

(9) Final rept.,

(10) BY JOHN E. HOLMES

STRATEGIC SYSTEMS DEPARTMENT

(11) 1 MARCH 1979

(12) 130p.

Approved for public release, distribution unlimited.

NSWC

NAVAL SURFACE WEAPONS CENTER

Dahlgren, Virginia 22448 • Silver Spring, Maryland 20910

391596

79 09 14 074

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NSWC/WOL TR 78-59	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MODIFLY: A Modular Multi-Degree-of-Freedom Trajectory Program		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John E. Holmes		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Surface Weapons Center White Oak Laboratory Silver Spring, MD 20910		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NIF; 0; 0; K80LA
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE March 1979 13. NUMBER OF PAGES 121
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Trajectory
Six-Degree-of-Freedom
Flight Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

MODIFLY is a modular trajectory simulation computer program which was written so as to be effective, efficient, and easily modified. The program was designed primarily for the simulation of typical autonomous guided missiles in which roughly equal consideration is given to the simulations of the seeker, guidance, autopilot, controls, and aerodynamics. It was written in FORTRAN IV language for use on a CDC 6500 computer and uses overlay files and a library edit routine.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SUMMARY

MODIFLY is a modular trajectory simulation computer program which was written so as to be effective, efficient, and easily modified. The program was designed primarily for the simulation of typical autonomous guided missiles in which roughly equal consideration is given to the simulation of the seeker, guidance, autopilot, controls, and aerodynamics. It was written in FORTRAN IV language for use on a CDC 6500 computer and uses overlay files and a library edit routine.

H. P. Caster

H. P. CASTER
By direction

CONTENTS

ILLUSTRATIONS

TABLE

INTRODUCTION

This program was written in order to accomplish two specific tasks. The first was to decrease costs and the second was to increase efficiencies in the simulation of the flight of any vehicle that moves above the earth's surface. These goals were attained by preparing a program which allowed for the condensing of several older NSWC trajectory programs into as small a package as possible in order to eliminate the excessive duplication of trajectory programs and, more importantly, to eliminate the excessive time consumed by the users in the maintenance of familiarity with each of the different programs.

This modular program consists of two main sections. The first, containing the executive routines, provides all of the control logic for the program from the specification of input data clear through to the final calculated results. Standardized, general formats are provided for the inclusion of all data. All necessary standard mathematical operations are coded and included, including means for the numerical integration of up to 28 differential equations, as well as standard generalized formats for the printing of the trajectory results.

The second section is written so that each user can select or program individual modules that meet his particular vehicle requirements. The program has been written in such a manner that it can be used to simulate any type of flight in the atmosphere -- including the simulation of guided vehicles from simple 3 DOF particle trajectories to maneuvering 6 DOF simulations of air-to-air missiles with proportional navigation or maneuvering re-entry bodies flying along evasive trajectories. Several basic modules as well as some specific modules have been written and are included in this report for the aid of the user. This program is efficient and easily modified by the user so that he can use it from the original conception of a system and its preliminary design through to its final flight evaluation.

EXECUTIVE ROUTINES

The primary functions of the executive routines are to control the program flow, establish standardized formats for the insertion of data and modules into the program, and provide for the economic storage of parameters and their use. The flow logic of the program is shown in Figure 1. In order to minimize storage, the program was coded using overlays. The individual subroutines included in each overlay level are shown in Figure 2 and described in the following section.

DESCRIPTION OF ROUTINES.

Program OV. This is the main, zero order overlay. Its function is to call the primary overlays; therefore, it controls the main flow of the program.

Subroutine ZERO. This routine zeros out the entire common storage array, Y(1) through Y(4940), and sets the following default values:

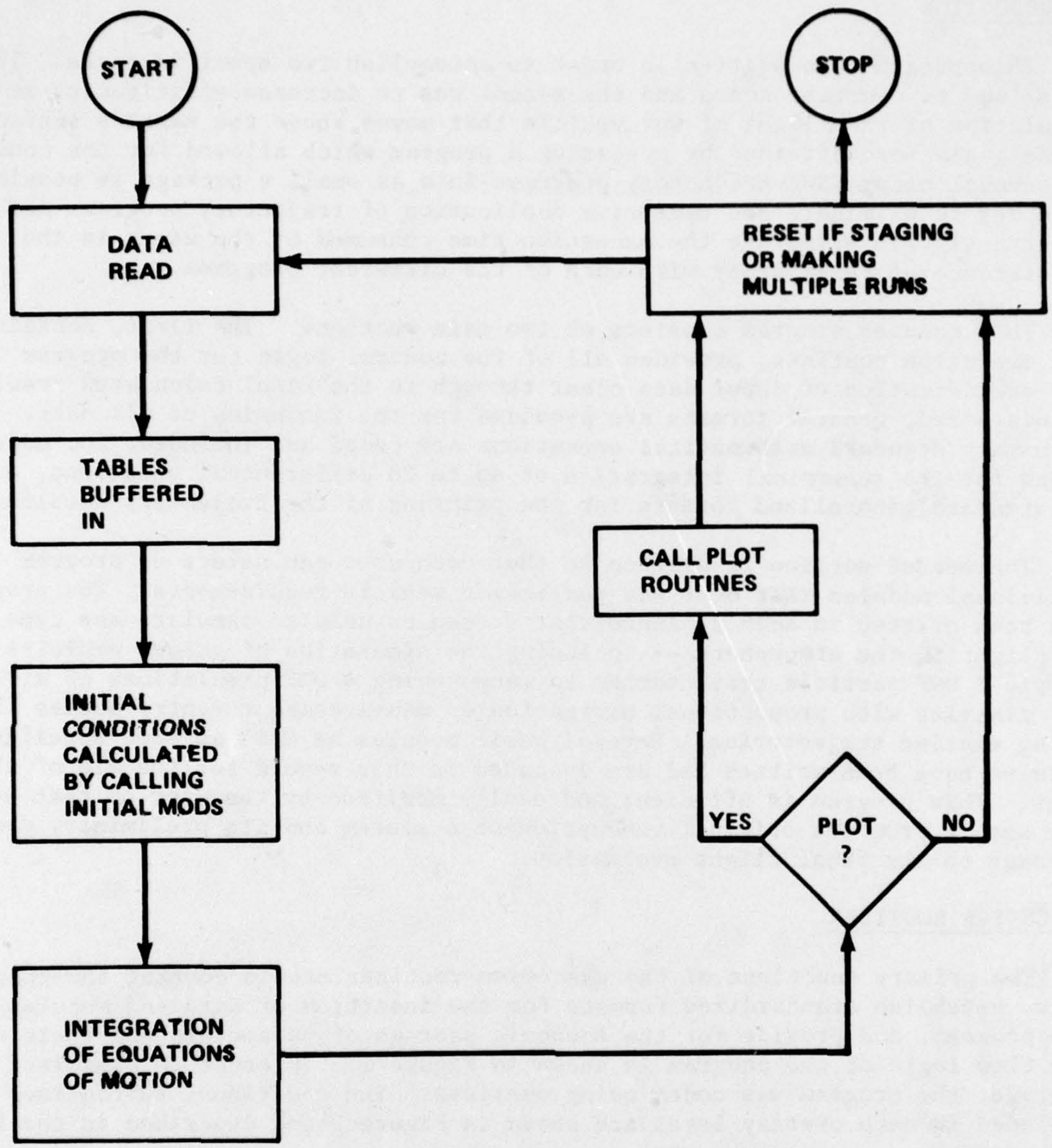


FIGURE 1 EXECUTIVE FLOW LOGIC

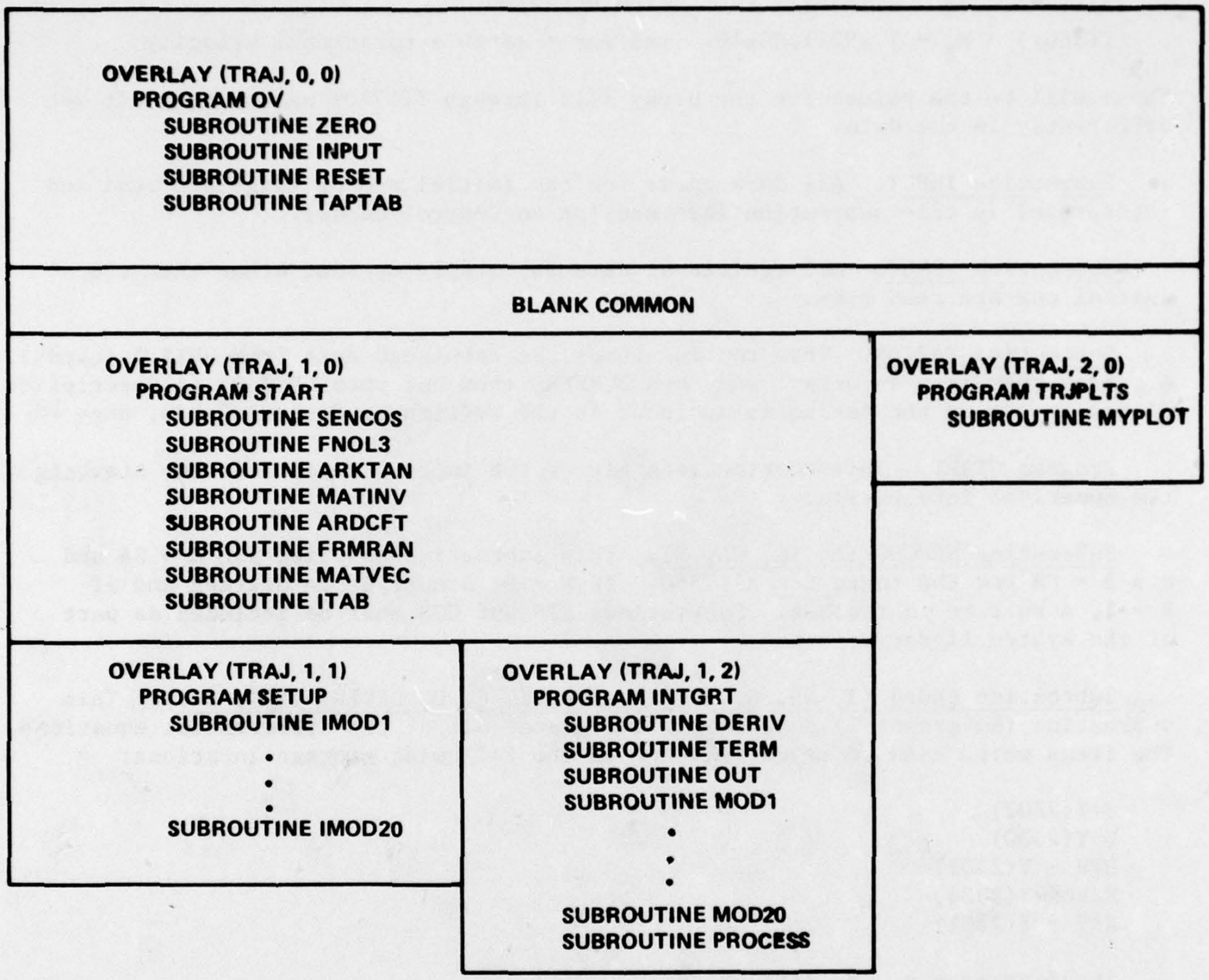


FIGURE 2 OVERLAY FILES

$$\left.\begin{array}{l} Y(2302) = J = 2 \\ Y(2304) = XNE = 0 \\ Y(2305) = MPR = 1 \\ Y(2306) = ERROR = 1 \end{array}\right\} \text{Integration Controls}$$

$Y(3000) = R_E = 20925631.$ ft = earth's radius

$Y(3001) = W_E = 7.29211508 \times 10^{-5}$ rad/sec = earth's rotational velocity

These will be the values for the array Y(1) through Y(4940) unless they are set differently in the data.

Subroutine INPUT. All data cards for the initial run or stage are read and interpreted in this subroutine (see section on Control Cards).

Subroutine RESET. All additional data for stages or runs other than the initial one are read here.

Subroutine TAPTAB. This routine reads the tabulated data from UNIT 5 (cards), arranges them into an array, and then BUFFERS them out onto UNIT 9. (A description of how to set up the tables is included in the section on Control Cards, page 13.)

Program START. This routine sets all of the initial conditions for starting the numerical integration.

Subroutine SENCOS (A, SA, CA, N). This subroutine supplies sin A = SA and cos A = CA for the range $0 \leq A \leq 360$. If N = 0, A must be in degrees and if N = 1, A must be in radians. Subroutines SIN and COS must be included as part of the system library.

Subroutine FNOL3 (J, NN, G, L, MPR, XNE, T, C, D, DERIV, TERM, OUT). This subroutine (Reference 1) numerically integrates all of the differential equations. The items which need to be defined are in the following storage locations:

J=Y(2302)
G=Y(2300)
MPR = Y(2305)
ERROR=Y(2306)
XNE = Y(2301)

These parameters are defined as:

J: (INPUT, INTEGER)

This parameter indicates the integration method.

---

[1] Ferguson, R. E. and Orlow, T. A., "FNOL3, A Computer Program to Solve Ordinary Differential Equations," NOLTR 71-2, 1 Mar 1971

J = 1 Use Runge-Kutta method of integration to termination. Truncation errors are not calculated; the step size G is not adjustable.

J = 2 Use Runge-Kutta for the first three steps, then Adams-Moulton for the remainder of the interval of integration. Truncation errors are calculated. The step size is adjustable unless XNE = 0. If the step size is adjusted, new starting values are obtained through the Runge-Kutta method.

J = 3 Use Runge-Kutta throughout. The truncation errors are calculated; the step size is adjustable unless XNE = 0.

G: (INPUT, REAL)

This is the initial step size.

MPR: (INPUT, INTEGER)

This is the print frequency -- the number of integration cycles between printouts. If MPR = 0, then printing is determined by values assigned to Y(2998) and Y(2997), where Y(2998) is set equal to some running variable like T, C(1), D(1), etc.and Y(2997) is a constant interval in Y(2998) between printing cycles.

XNE: (INPUT, REAL)

This is the step size control. The step size is unchanged if the worst of all the errors lies within the window 10^{-XNE-3}, 10^{-XNE}. The step size is increased if the errors are all less than 10^{-XNE-3}. The step size is decreased if for some differential equation the error is greater than 10^{-XNE}. If ERROR < 0 and XNE $\neq$ 0., the automatic adjustment of the step size is a function of the absolute errors. If ERROR = 0. and XNE $\neq$ 0., the automatic adjustment of the step size is a function of the relative errors. If ERROR = $\varepsilon > 0$. and XNE $\neq$ 0., the automatic adjustment of the step size is a function of the relative errors where the relative errors are equal to the absolute errors divided by the maximum ERROR, $|C(I)|$). This option removes the possibility of using "small" functional values to compute relative error, otherwise this option is identical to the previous option and is to be preferred over it. If XNE = 0., the step size G is not adjustable. The other parameters are either set internally in the program or are defined in the section on Control Cards.

Subroutine ARKTAN (A, B, C, N). This subroutine calculates arctangents defined as $C = \tan^{-1}(A/B)$. If N = 0, C is in degrees and if N = 1, C is in radians. The range of C is: $-180 \leq C \leq +180$. If A = 0, B < 0. C = -180; and if B > 0. then C = +180. Subroutine ATAN must be in the system library.

Subroutine MATINV (A, B, C). This subroutine computes the transpose (B) and inverse (C) of the (3, 3) matrix (A). If the determinant of A is zero, neither B or C is calculated; instead, a comment is printed and control is returned to the calling program.

Subroutine ARDCFT (H, P, T, D, C, G). The earth's atmospheric properties (Reference 2) are supplied by this subroutine up to an altitude of 10^6 feet. Entering with the altitude (H, ft.) the pressure (P), temperature (T), density (D), speed of sound (C) and acceleration due to gravity (G) are given ratioed to their corresponding sea level values.

Subroutine FRMRAN (TABLE, NUM, MFNC, U, A). This is a linear interpolation routine which extracts tabulated data from TABLE, and then with the NUM independent variables U_1, U_2, ... U_{NUM} it linearly interpolates or extrapolates $2^{NUM} - 1$ times and supplies the MFNC values of the functions A.

Subroutine MATVEC (A, B, C, N). Products of matrices, whose orders are (3, 3) and (3, 1) are computed by this subroutine. When:

N = 0, $C = AB$*
N = 1, $C = A^TB$*
N = 2, $C = AB$
N = 3, $C = A^TB$
N = 4, $C = AB^T$
N = 5, $C = A^TB^T$

The A, B, and C arrays are stored, column-wise, starting at the left. The symbol * indicates that B, in these cases, is a (3, 1) array. In all other cases A, B, C are (3, 3) arrays.

Subroutine ITAB (NTAB, N, U, V). This routine selects the table designated by NTAB, which is the numerical location of the table in the KTAB array, and for N independent variables of U calls FRMRAN for the linear interpolation of the function V.

Program SETUP. This program calls the initial modules IMOD1 through IMOD20 as designated by the code 1 control cards.

[2] U.S. Standard Atmosphere, 1969 (NASA, Dec 1969, Washington, DC)

Subroutines IMOD1 through IMOD20. These dummy subroutines are included so that users can substitute their own subroutines for calculating any initial conditions that may be needed before the numerical integration is started.

Program INTGRT. This program sets up the integration controls and calls FNOL3 for the numerical integration of the equations of motion.

Subroutine DERIV. This subroutine calls the appropriate modules, MOD1 through MOD20, as designated by the code 1 control cards.

Subroutine TERM. The termination conditions (code 4 control cards) are checked and if any one of them has been met, the integration is terminated.

Subroutine OUT. The output is prepared and printed here based on the information included on the code 2 control cards.

Subroutines MOD1 through MOD20. These are dummy subroutines. The user should substitute his own subroutines for the dummy ones. These subroutines are to contain all of the definitions for all of the differential equations.

Subroutine PROCESS. Again, this is a dummy subroutine which can be replaced by the user. Any accessary calculations that are not needed for the integration of the differential equations are usually included in this subroutine. The subroutine is called in subroutine OUT everytime that the print conditions have been met.

Program TRJPLTS. This program contains the calls for plotting any of the data designated on the code 5 control cards.

Subroutine MYPLOT. This is a dummy subroutine. If the user wishes to plot any variables he must substitute his own MYPLOT subroutine containing his own GOULD or equivalent plot calls.

A FORTRAN listing of these executive routines has been included in Appendix A.

STORAGE ALLOCATION. As mentioned earlier, the program has been coded using overlay files in order to minimize the machine memory required. The amount of storage needed for execution will vary according to the size of the particular modules used as well as by the size of the arrays that are to be plotted and dimensioned in subroutine MYPLOT. Generally, on WOL's CDC 6500, the user has needed around $45000_{(8)}$ locations. The maximum has been on the order of $63000_{(8)}$ locations.

All of the parameters stored in the program have been placed into an array dimensioned Y(4940). The first 2299 locations have been allocated to trajectory parameters. These are available for coding in the modules. The locations Y(2300) through Y(4940) are utilized in the executive routines and as such, are not available to the user. As an aid to keeping track of the parameters, the Y array has been broken into several parts. It is not absolutely necessary for the user to retain this designation in his modules; but, it is a great aid to keeping all modules interchangeable. This Y array breakdown is shown in Figure 3. The fixed storage assignments are listed in Appendix B.

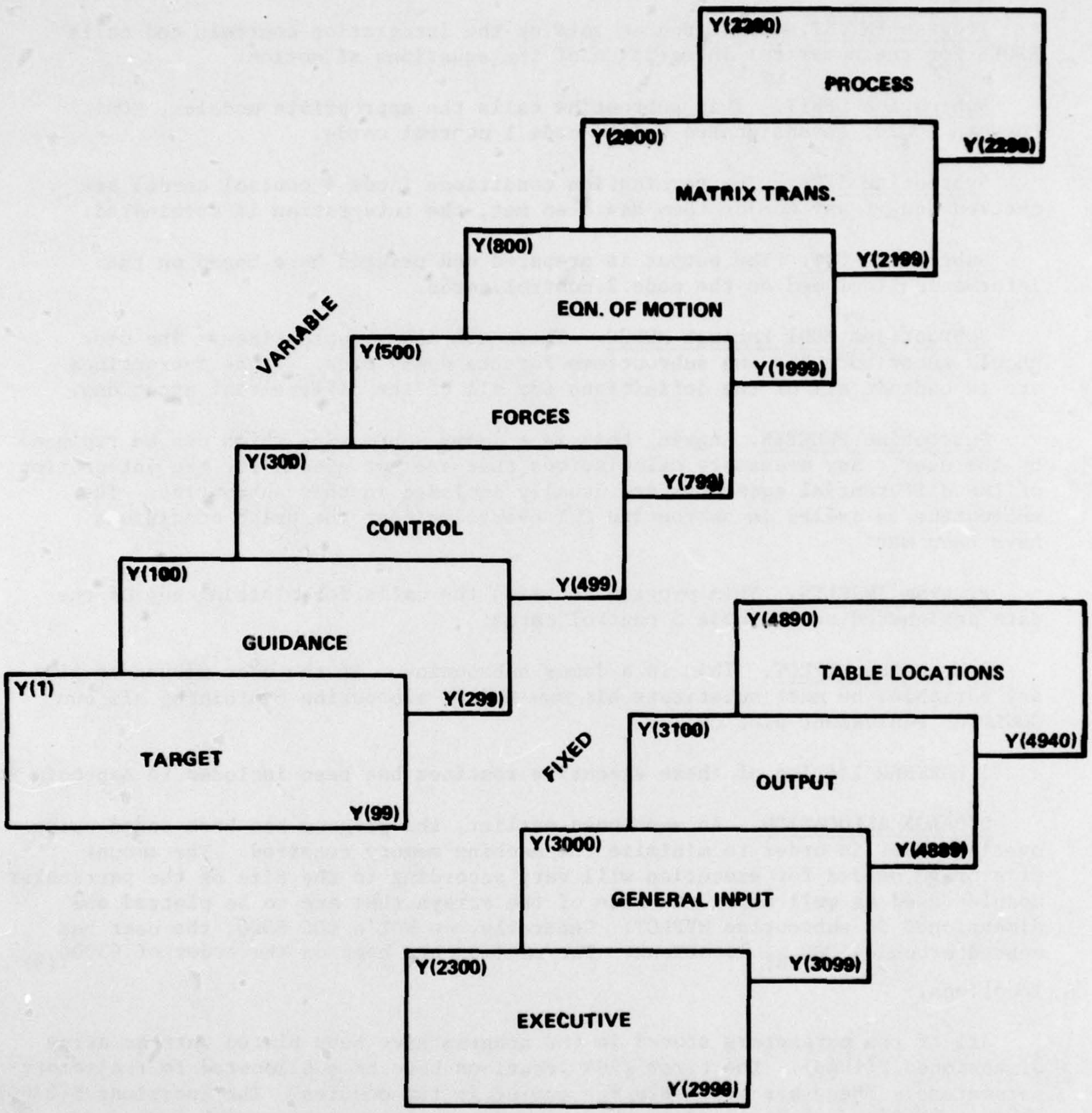


FIGURE 3 Y ARRAY STORAGE

CONTROL CARDS. All data necessary for running this program are entered into the program through the use of standardized formats. Each piece of information, except for tables, is entered on a separate control card. The particular use for each data card is determined by the code punched in the first two columns of the card. The codes, formats, and types of information are tabulated in Table 1 on page 14 and explained in more detail below.

Code 0. Each data deck must contain this card as the first card in the deck. The users title located in columns 3 through 72 will appear in the heading at the top of each page of printout.

Code 1. These are the modules to be used for this particular run or stage. These cards must be read in the order you wish the module to be called. It is not necessary that the mod numbers be in sequence.

Code 2. These are the variables to be listed in the output. The data will be listed in columns with the first column always being time. The first 15 variables will be listed on the first page and the next 15 on a separate page. The number of variables listed can be any number from 1 through 30; but, note that if you print the results of no more than 15 variables you will save paper, time, and money. The columns will be printed in the order that the code 2 control cards appear in the deck. Each code 2 control card is to contain the location of the parameter in the Y array, the heading that you wish to appear at the top of the column, and the format of the parameter. The maximum width of each column is 8 spaces but you can place the decimal depending on what is being listed. If the format is left off the card, the default is F8.0.

Code 3. These cards contain the initial values of any parameters in the program. They may be in any order but the total number for all stages must not exceed 200.

Code 4. This code identifies the termination conditions for the numerical integrations. The program will stop whenever any parameter in the Y array designated as a stop variable goes outside of the lower or upper limits as set by this code. As many as 10 termination conditions may be set for any complete trajectory run.

Code 5. Any variable in the Y array that is to be plotted (other than time) must be designated with a code 5 control card. Whenever any code 5 control card (up to a maximum of 10) appears in the deck, subroutine MYPLOT will be called and the users plot options will be performed.

Codes 6 and 7. A maximum of 28 differential equations can be designated with these codes. The code 6 designates the dependent variables and the code 7 their derivatives. The variable locations in the "C" and "D" arrays must correspond; i.e., the first variable in the "D" array must be the derivative of the first variable in the "C" array.

TABLE 1 CONTROL CARDS

CARD COLUMN	1-2	3 – 8	9 – 22	23 – 36	37 – 44 – 46	47 – 53	54 – 63	64 – 72
FORMAT	I2	I6	E14.6	E14.6	A10	A7	A10	A9
VARIABLE NAME	IR	IN	VAR	VARR	HED1	HED2	HED3	
TITLE	0	IDENTIFICATION, 7A10						
MODULES	1	MOD NUMBER					USERS IDENT.	
OUTPUT PARAMETERS	2	LOCATION IN Y ARRAY			HEADING FOR PRINTOUT 44		FORMAT F8?	
DATA	3	LOCATION IN Y ARRAY	NUMERICAL VALUE				USERS IDENT.	
TERMINATION CONDITIONS	4	LOCATION IN Y ARRAY	LOWER BOUNDARY	UPPER BOUNDARY			USERS IDENT.	
PLOT PARAMETERS	5	LOCATION IN Y ARRAY					USERS IDENT.	
DEPENDENT VARIABLES	6	LOCATION IN Y ARRAY	LOCATION IN C ARRAY				USERS IDENT.	
DERIVATIVES	7	LOCATION IN Y ARRAY	LOCATION IN D ARRAY				USERS IDENT.	
TABULATED VARIABLES	8	TABLE NUMBER	KTAB ARRAY INDEX					
TABLES TO FOLLOW	9							
END OF DATA FOR RUN	10							
BEGIN NEXT STAGE	11	IDENTIFICATION, 7A10						
STOP, END OF DATA	99							

Code 8. These control cards are used to indicate where the tabulated functions appear in the deck of tables. When the modules are coded, an index in the KTAB array is assigned to each tabulated function. For example, maybe the axial force coefficient was coded as having index number 12 in the KTAB array. For this particular run, the axial force coefficient table that you wish to use may be the fourth table in your deck of tables; therefore, IN equals 4 and VAR equals 12.

Code 9. A card with the number 9 in column 2 must proceed the deck of tables. It causes the program to read the following cards as tabulated data. The tables should be arranged as follows:

	9	
Table 1	Title card	(FORMAT 7A10)
	Control card	(FORMAT 14I5)
	Listing of independent variables	(FORMAT 6E12.7)
	Listing of values of dependent function	(FORMAT 6E12.7)
Table 2	Title card	
	Control card	
	Listing of independent variables	
	Listing of values of dependent function	
Table I (I=3, 49)	repeat for as many tables as needed	
	BLANK CARD	at end of tables
	BLANK CARD	

In this program the tabulated functions are functions of 1, 2, or 3 variables, with each function located in a separate table. The tabulation of a function of three variables would be as follows:

a. Control Card

$L, N, M, n_1, n_2, \ldots n_N$ (FORMAT 14I5) where:

$L = 0$	all cases
$N = 3$	number of independent variables
$M = 1$	each table contains only 1 function
n_1, n_2, n_3	numbers of values of each independent variable for which values of the function are tabulated.

b. <u>Listing of values of independent variables for which function is tabulated</u>

On the first card(s) the n_1 values of the first independent variable are listed. On succeeding cards the n_2 and n_3 values of the second and third independent variables, respectively, are listed.

The following restrictions apply: Values of two different independent variables may not appear on the same card. At least two values of each independent variable must be listed; and, all values must be distinct and must be listed in ascending order. The format for these cards is 6E12.7.

c. <u>Listing of values of dependent function</u>

The three independent variables are designated N_1, N_2, and N_3. The number of values of each of these variables is n_1, n_2, and n_3, respectively. A block of data contains those values of the function for all values of N_3 listed, and for one particular value of N_1 and N_2. The first block corresponds to the first value of N_1 and N_2 listed; the second block corresponds to the first value of N_1 and the second N_2, etc. These blocks are repeated until a set of blocks for the first value of N_1 and all values of N_2 have been presented. Sets for the remaining values of N_1 follow until the table is completed. As a check there are n_1 sets, $n_1 \times n_2$ blocks and $n_1 \times n_2 \times n_3$ distinct values of the function. The format for this tabulation is also 6E12.7.

<u>Code 10</u>. This card is to be placed at the end of the data for that particular run or stage of the run. When this card is read, the program will stop reading data cards and start integrating the equations of motion. When one of the termination conditions has been met, the program will stop integrating and start reading the next coded data card. At this point in time, the program still retains the initial conditions as read in at the beginning of the run as well as all of the values as last calculated when the termination conditions were met.

If you wish to stack runs, i.e., start another run with slightly different initial conditions, follow the code 10 card with a new code 0 title card, and then follow this with the necessary changes that you wish to make in the original code 3 data. The program will retain the original initial code 3 conditions except for those that you change here. You must include new code 1, 4, and 8 cards; i.e., tell the program which modules to use, new stop conditions, and which tables to use.

<u>Code 11</u>. This is a title card that is an indication to the program that staging is to take place. The general procedure is that the program will read additional data at this time. These data will then replace the values retained by the program when the last portion of the flight was terminated. This allows you to restart the calculations where you finished. The only code 3 data required are those that you wish to change at that point in the trajectory. You must include new code 1, 4, and 8 cards, i.e., tell the program which modules to use, which tables to use, and new stop conditions. Remember though, the total number of data variables, code 3 cards, must not exceed 200 and the total number of tables must not exceed 49 for all stacked runs or stages.

Code 99. This stops the program.

TRAJECTORY MODULES

The primary reason for writing this program was to build a program which could be utilized and changed by a wide variety of users without them having to spend an inordinate amount of time in learning and adopting the code. It was also envisioned that the program had to be of use to those conducting preliminary design studies (when only the basic fundamentals of the vehicle are known and the vehicle characteristics are constantly changing) as well as for those analyzing the flight mechanics of production systems. In order to refrain from writing a general purpose program that would cover as much detail as possible for all users, but satisfy none, it was decided to program the problem so that different modules, written for specific systems, could be selected or written and inserted by each user.

In order to keep each module as universal in its use as possible, it was necessary to break the system model into several parts and to minimize the linkage between each part. In general, the parts of a system can be divided as is shown in Figure 4.

The logic behind these parts was as follows. The only information that would be passed from the target module to the seeker module would be the target coordinates. The interface between the seeker and autopilot modules would usually consist of a maximum of three error signals. The interface between the autopilot and the force and moment modules would be the two or three control deflections and the only thing that would have to be passed over to the equations of motion are the three forces and three moments.

Note though that this arbitrary division of the modules while appearing to be logical is not permanently locked into the code. For any simulation as many as 20 modules can be used and the quantities exchanged between modules can be chosen at the whim of the programmer. These are just suggested modules which will aid in the exchange of modules among all of the users.

The direction cosine matrix is generally defined in the section called "Matrix Transformations." When using this program for 6DOF simulations the position of the body principal axes with respect to the inertial axes is generally defined by integrating the elements of the direction cosine matrix. These elements and their derivatives are defined in this module as well as other transformation matrices such as the inertial to local axes transformation, and the local to principal axes transformation. Generally, one of two modules will take care of the matrix transformations. One of these is for 3DOF and the other for 6DOF. These have been coded and are included in this report as MOD1 and MOD4.

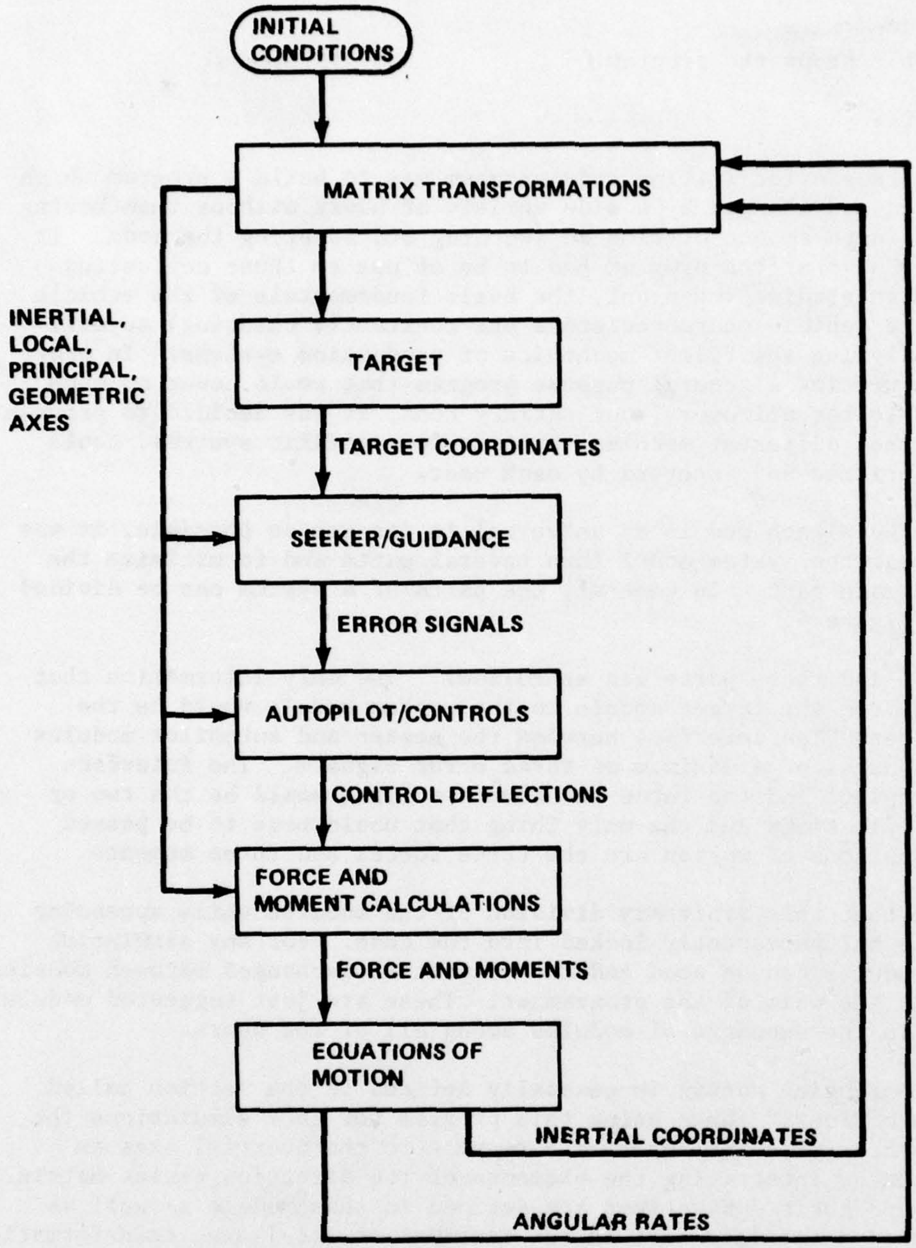


FIGURE 4 MODULE LINKAGE

The target module is also of a general nature. It allows for a target to either remain fixed with respect to the earth or to fly at constant velocity over the earth. This general module is also included in this report as MOD5. If the user desires any particular form of maneuvering target he can easily insert his own module.

The seeker and autopilot are more system oriented; therefore, they are generally coded for a particular system. Some examples are included in this report as MOD6 and MOD7.

Likewise, the forces and moments generated on the vehicle are dependent on what kind of control system, etc., the vehicle has. It could have canard controls, aft fins, swivel nozzle, etc. A sample has been included as MOD8.

The equations of motion are of a more general nature; therefore, examples for modules for both 3DOF and 6DOF are included in this report as MODs 14 and 9.

Remember, you only need to call the modules which pertain to your case. In early studies of a preliminary design, you may only need one or two very simple modules. As the system is developed you can then expand and add to your module package. You can also take advantage of another's modules that have been developed if all parties maintained the general interchangeability features as outlined in this report.

AXIS SYSTEMS AND TRANSFORMATIONS. In general, only four different axis systems are needed. These are inertial, local, principal, and geometric axis. These can be used to specify orientations of the missile with respect to the target and with respect to the earth.

Inertial. The inertial axes are defined as follows. The origin is at the center of the earth and the X_R axis goes through the north pole and the Y_R and Z_R axes go through the equator as is shown in Figure 5. These axes are the ones in which the force equations are integrated.

Local. The local axes, labeled by the subscript "L" have their origin on the earth's surface right below the vehicle. The Z_L axis is perpendicular to a tangent plane located in the earth's surface and passes up through the vehicle's center of gravity. The X_L axis lies in the tangent plane and always points north. The origin of the local axes are defined by the longitude, τ_R, and latitude, ψ_R, of the vehicle. These relationships are shown in Figure 5.

The transformation matrix for transferring a vector in the inertial system to one in the local axes is designated $[\ell_{RL}]$ where,

$$\begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}_L = \begin{bmatrix} \ell_{RL} \end{bmatrix} \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}_R$$

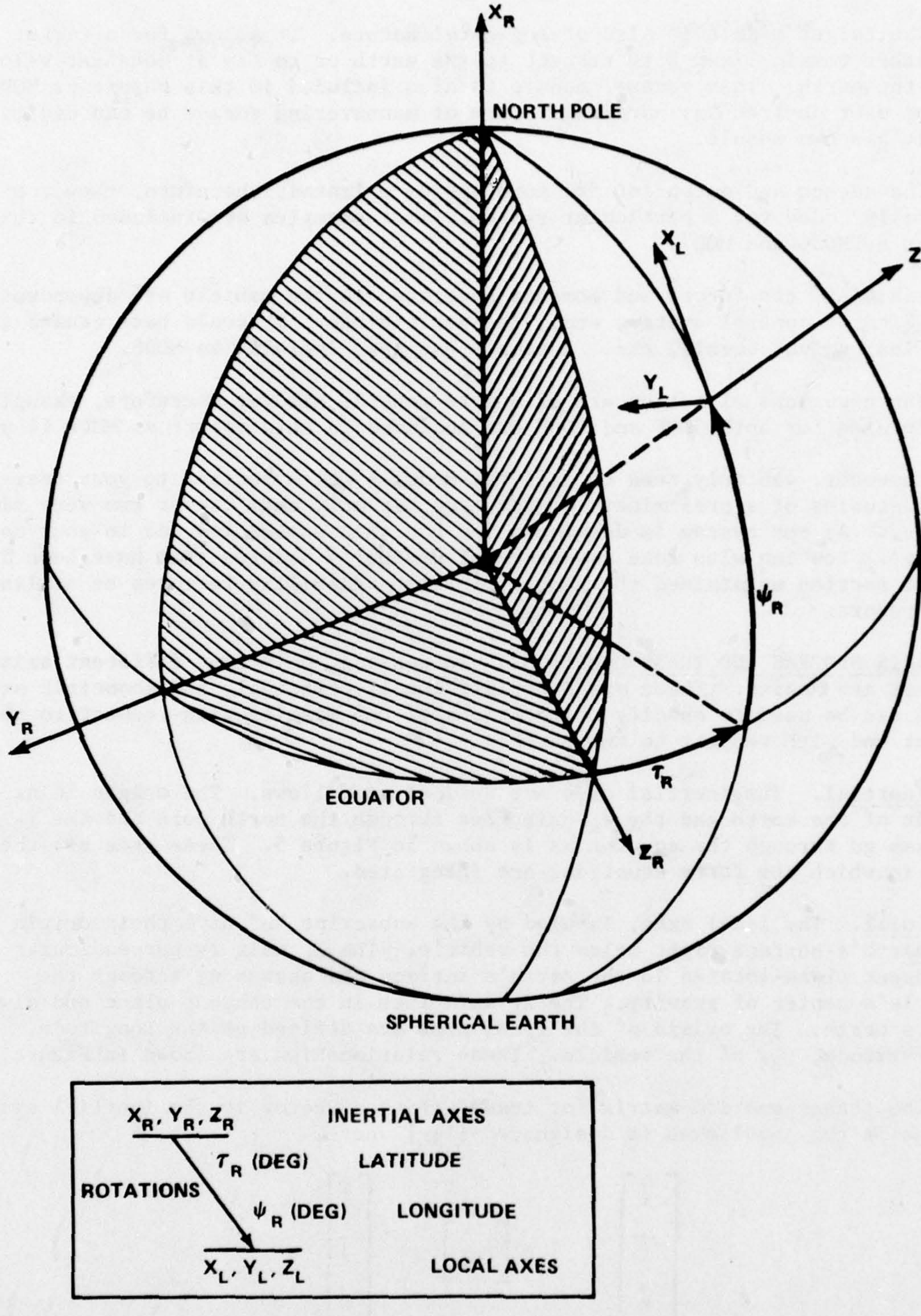


FIGURE 5 INERTIAL AND LOCAL AXES

and

$$\left[\ell_{RL}\right] = \begin{bmatrix} \cos\psi_R & \sin\psi_R \sin\tau_R & -\sin\psi_R \cos\tau_R \\ 0 & \cos\tau_R & \sin\tau_R \\ \sin\psi_R & -\cos\psi_R \sin\tau_R & \cos\psi_R \cos\tau_R \end{bmatrix}$$

The winds are generally defined with respect to these local axes as shown in Figure 8.

Principal Axes. These are the principal axes of the vehicle, i.e., their origin is at the center of mass and they consist of a set of cartesian axes for which the inertia tensor will be a diagonal (see Figures 6 and 7). The initial orientation of the principal axes with respect to the local axes is defined by the three angles γ_M, ε_M, and ϕ_M (in that order) where,

$$\begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}_P = \left[\ell_{LP}\right] \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}_L$$

and

$$\left[\ell_{LP}\right] = \begin{bmatrix} \cos\varepsilon_M \cos\gamma_M & -\cos\varepsilon_M \sin\gamma_M & \sin\varepsilon_M \\ \cos\phi_M \sin\gamma_M - \sin\phi_M \sin\varepsilon_M \cos\gamma_M & \cos\phi_M \cos\gamma_M + \sin\varepsilon_M \sin\gamma_M & \sin\phi_M \cos\varepsilon_M \\ -\sin\phi_M \sin\gamma_M - \cos\phi_M \sin\varepsilon_M \cos\gamma_M & -\sin\phi_M \cos\gamma_M + \cos\phi_M \sin\varepsilon_M \sin\gamma_M & \cos\phi_M \cos\varepsilon_M \end{bmatrix}$$

The means of arriving at this matrix after the initial time is defined and explained in MOD4.

Geometric Axes. These are a set of orthogonal axes which are defined for maximum convenience in expressing the vehicle aerodynamics. They will generally define a plane of symmetry for the vehicle external configuration and their origin will be located at the moment reference center. The relationship between these axes and the principal axes are shown in Figure 7. The center-of-gravity (c.g.) of the vehicle is defined with respect to the origin of the geometric axes by the lateral transformations X_{cg}, Y_{cg} and Z_{cg}. The angular orientation is expressed through the rotations θ_G, ψ_G, and ϕ_G. These transformations are performed in the force and moment module, MOD8. The transformation matrix is defined as,

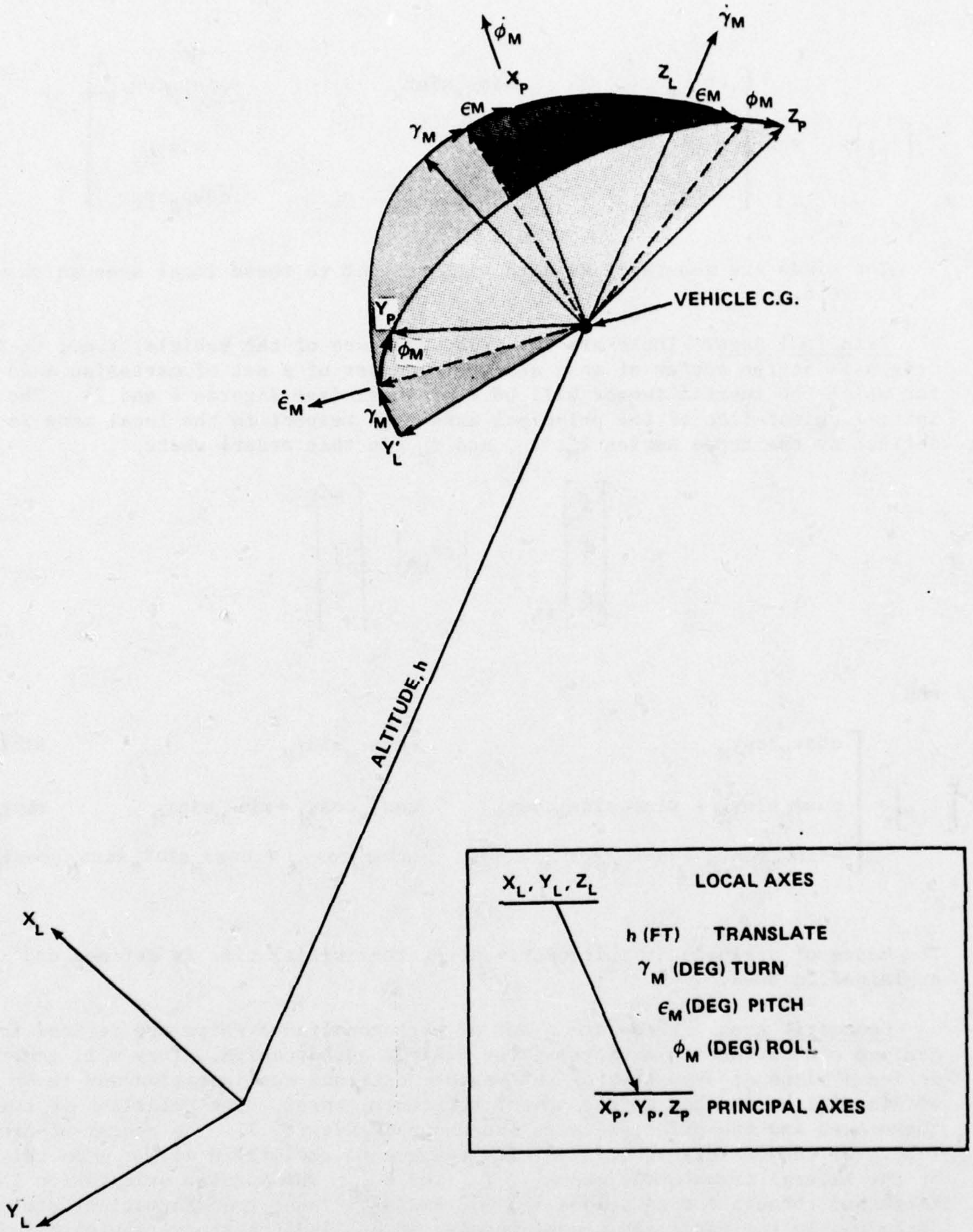


FIGURE 6 LOCAL AND PRINCIPAL AXES

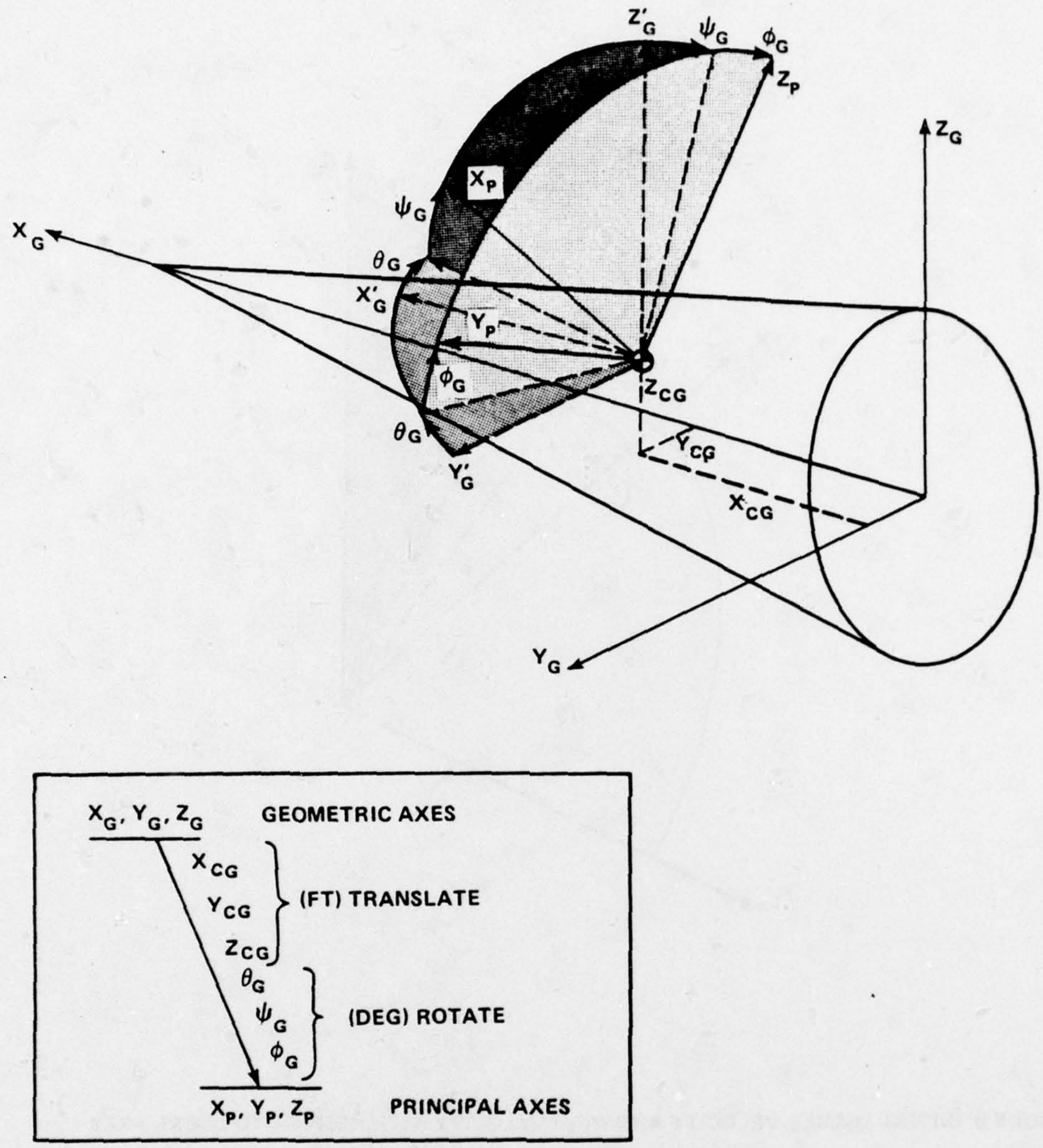


FIGURE 7 PRINCIPAL AND GEOMETRIC AXES

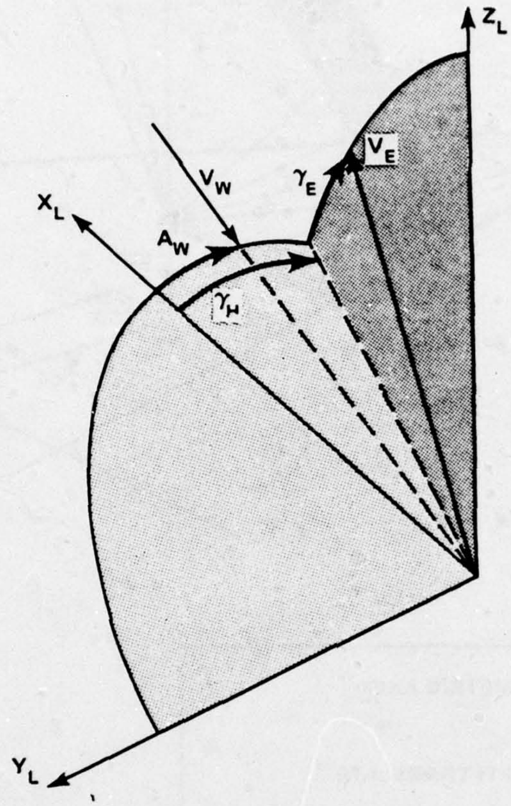


FIGURE 8 INITIAL MISSILE VELOCITY AND WIND VELOCITY WITH RESPECT TO LOCAL AXES

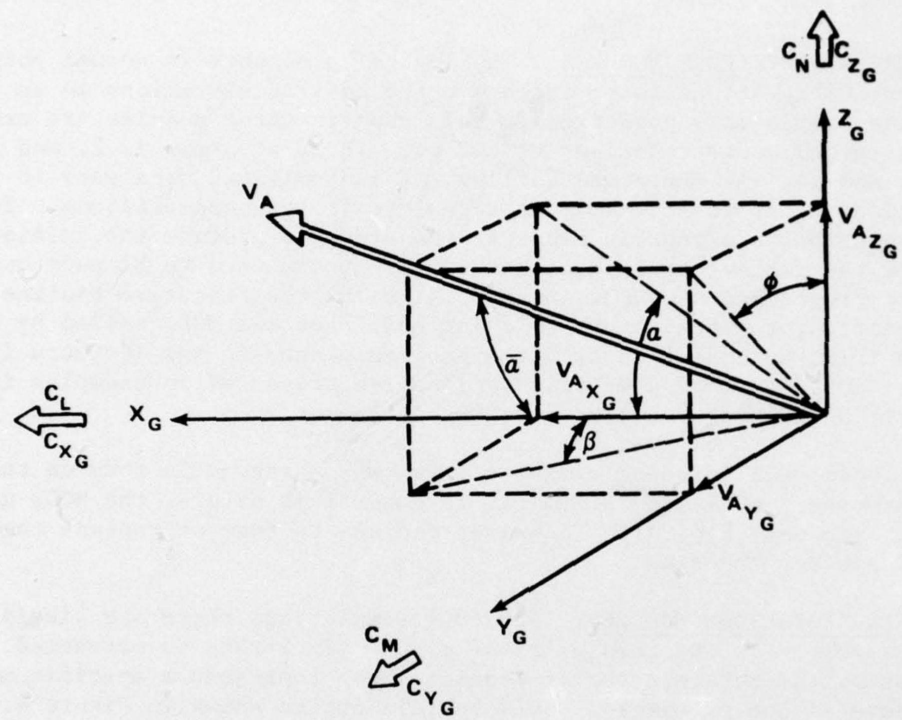


FIGURE 9 AERODYNAMIC COEFFICIENTS AND ANGLES DEFINED IN GEOMETRIC AXES

$$\left[\ell_{GP}\right] = \begin{bmatrix} \cos\psi_G \cos\ _G & -\cos\psi_G \sin\ _G & \sin\psi_G \\ \cos\phi_G \sin\theta_G - \sin\phi_G \sin\psi_G \cos\theta_G & \cos\phi_G \cos\theta_G + \sin\phi_G \sin\psi_G \sin\theta_G & \sin\phi_G \cos\psi_G \\ -\sin\phi_G \sin\theta_G - \cos\phi_G \sin\psi_G \cos\theta_G & -\sin\psi_G \cos\theta_G + \cos\phi_G \sin\psi_G \sin\theta_G & \cos\phi_G \cos\psi_G \end{bmatrix}$$

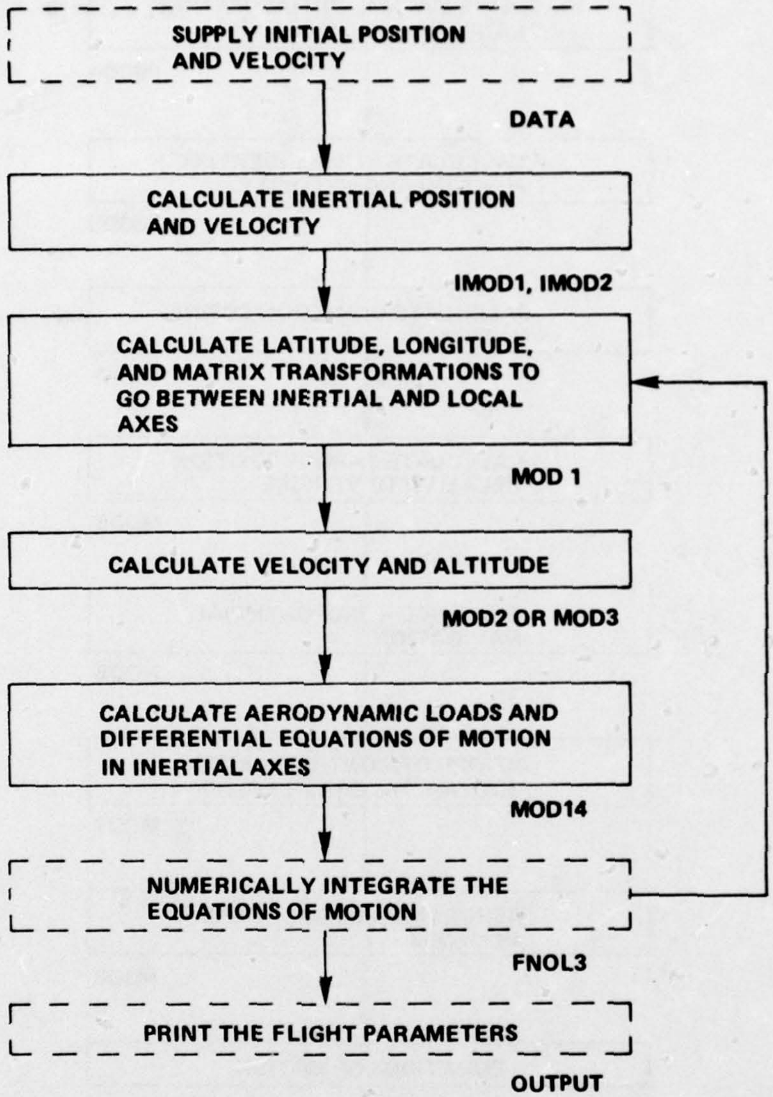
Aerodynamic coefficients and orientations of the vehicle with respect to the flow are generally defined with respect to these geometric axes as is shown in Figure 9.

THREE-DEGREES-OF-FREEDOM MODULES. The logical procedure in normal point-mass trajectory calculations is to integrate the body accelerations in an inertial system. In the sample case presented in this report, three modules are used to define the second order equations of motion. These are MODs 1, 2, and 14 or MODs 1, 3, and 14. As mentioned earlier, it is sometimes necessary to perform some calculations prior to starting the actual trajectory calculations. These types of calculations are usually necessary in order to provide the initial conditions for the differential equations. Since these need to be made only once, they are programmed as IMODs and are called by the executive routine prior to actually starting the trajectory calculations. For all MODs called by the program (identified by code 1 cards), the program checks to see if there is a corresponding IMOD. For the 3DOF calculations (as presented in examples included in this report) an IMOD is required for MODs 1, 2, and 3.

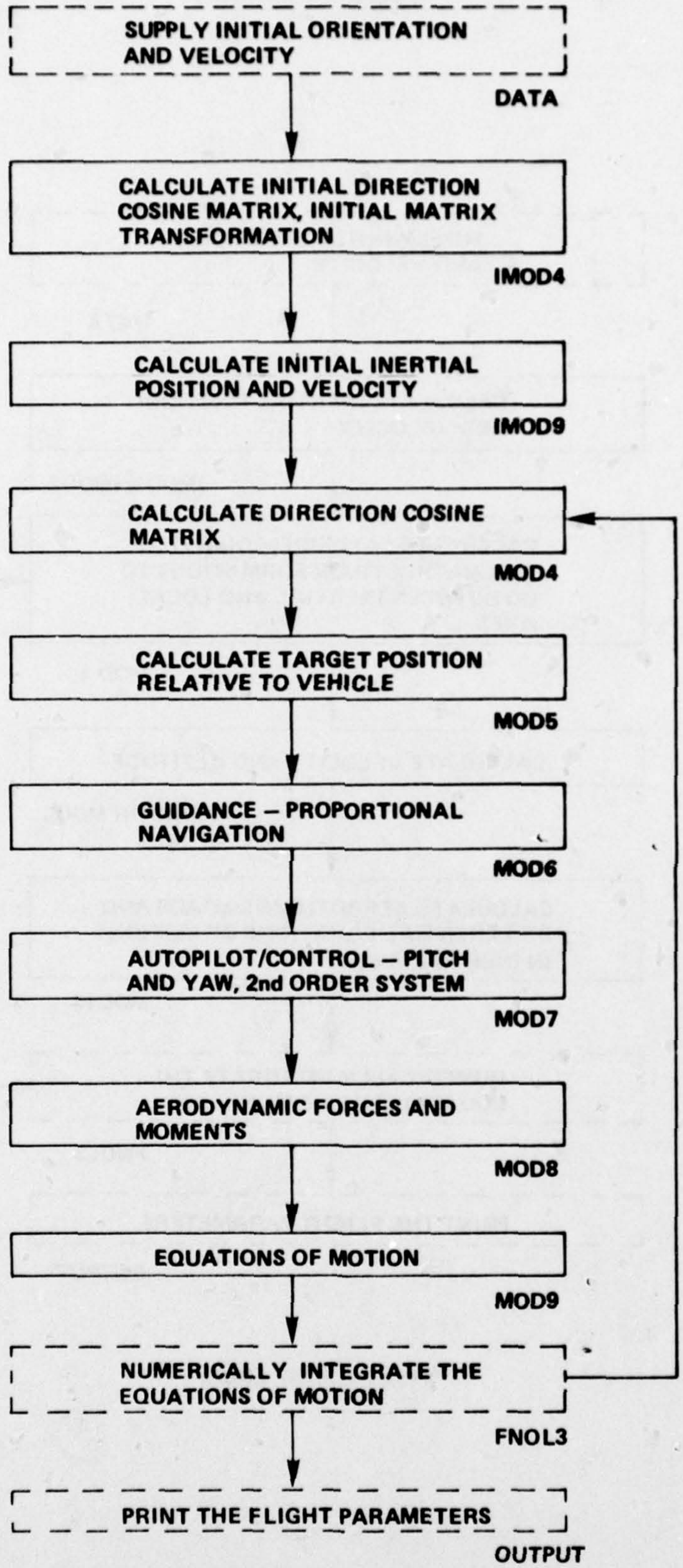
The flow logic used in these examples is shown in schematic form on the next page. Remember, while the schematic is general in nature, the MODs used in this report are only typical. Each user can add to them or replace them with those of his own choosing.

SIX-DEGREES-OF-FREEDOM MODULES. For 6DOF simulations there are likely to be six modules or more. If the complexity of a 6DOF simulation is warranted, it is usually necessary to prepare a set of modules which represent a specific missile system. In general the simulation would be laid out as shown in Figure 4. A group of typical modules have been selected and enclosed with this report in order to aid the user in the compilations of modules for his own simulation. The logic of these sample modules is shown on the following page.

IMOD4 and MOD4 are used to calculate the direction cosine matrix. It is unlikely that any changes or additions would have to be made to these modules since they deal with the mechanics of the situation, not a particular piece of hardware. MOD5 is used to calculate the location of the target. Here again, unless a particular maneuver of the target is to be programmed, it is unlikely that changes would be made to this module. IMOD9 and MOD9 are used to calculate the equations of motion of a vehicle for which the cross products of inertia are zero. MOD19 should be used in place of MOD9 if the cross products of inertia are not zero.



TYPICAL 3DOF LOGIC



TYPICAL 6DOF LOGIC

PROCESS. This routine is used to calculate any quantities that are desired for output but which are not necessary for the running of the program. An example of the type of things which can be calculated here are the longitudinal range along the equator, R_{τ_E}, and the latitudinal range, R_{ψ_E}. These are calculated as:

$$R_{\tau_E} = R_E[\tau_R - \omega_E(t - t_i)] - R_{\tau_{Ei}}$$

$$R_{\psi_E} = R_E \psi_R - R_{\psi_{Ei}}$$

A calculation of the total distance traveled over the earth's curved surface (projection of vehicle's path on the earth's surface) is calculated in the following manner.

$$\tau_E = \tau_R - \omega_E t$$

$$\Delta X^2 = [R_E \sin \psi_{RL} - R_E \sin \psi_R]^2$$

$$\Delta Y^2 = [-R_E \cos \psi_{RL} \sin \tau_E + R_E \cos \psi_R \sin \tau_E]^2$$

$$\Delta Z^2 = [R_E \cos \psi_{RL} \cos \tau_E - R_E \cos \psi_R \cos \tau_E]^2$$

where ψ_{RL} and τ_E are the last latitude and longitude calculated (last time program executed subroutine PROCESS). The increment of the chord of travel over a segment of the earth's surface is then expressed as,

$$\Delta C = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}$$

and the angle subtended by the chord as

$$\theta_c = 2\sin^{-1}\left(\frac{\Delta C}{2R_E}\right)$$

The increment of surface traversed is then equal to

$$\Delta R_S = R_E \theta$$

and

$$R_S = R_S + \Delta R_S$$

is the total distance traveled.

Any similar calculations can be made and added to PROCESS. See Appendix P for its current form.

SETUP PROCEDURE

The required parameters in the setup deck will depend on what modules are in use; but, the general procedure and types of cards are usually the same. The following is a generalized listing of the most used setup and control cards. The general procedure is to prepare the deck as follows:

1. Title card, code 0.

2. Select the appropriate modules for your simulation and arrange the code 1 control cards in the order that the modules are to be called.

3. Provide the data cards, code 3, required by the executive routines. These are mostly integration controls and general physical descriptors. These are located in the executive and generalized input sections of the Y array, locations Y(2300) - Y(3099). See Appendix B for specific, required parameters.

4. Determine what the termination conditions are and set their values with code 4 control cards.

5. Specify all of the remaining initial conditions required by the modules with code 3 cards. These are parameters which generally have to do with defining the initial attitude of the vehicle.

   a. Locate the local axes by giving the longitude, τ_R (deg) in Y(3014) and the latitude, ψ_R (deg) in Y(3015).

   b. Define the initial altitude, h(ft) in Y(3013).

   c. Locate the principal axes with respect to the local axes by specifying,

$$\left.\begin{array}{lll} \gamma_M(\text{deg}) & \text{in} & Y(2066) \\ \varepsilon_M(\text{deg}) & \text{in} & Y(2067) \\ \phi_M(\text{deg}) & \text{in} & Y(2068) \end{array}\right\} \quad \text{IMOD4}$$

   d. Define the initial velocity with respect to the local axes.

$$\left.\begin{array}{lll} V_E(\text{fps}) & \text{in} & Y(610) \\ \gamma_E(\text{deg}) & \text{in} & Y(2209) \\ \gamma_H(\text{deg}) & \text{in} & Y(2208) \end{array}\right\} \quad \begin{array}{l}\text{IMOD2} \\ \text{MODs 2, 3}\end{array}$$

e. Define the angular orientation of the principal axes with respect to the geometric axes that you are using.

X_{CG}(ft)	in Y(3006)	
Y_{CG}(ft)	in Y(3007)	
Z_{CG}(ft)	in Y(3008)	MOD8
θ_G(deg)	in Y(3019)	
ψ_G(deg)	in Y(3020)	
ϕ_G(deg)	in Y(3021)	

f. Define all other constants required by the modules you are using.

6. Decide what parameters you want listed in the output and identify them, their titles, and formats on code 2 control cards.

7. List all of the dependent variables on code 6 control cards.

8. List all of the derivatives for the above dependent variables on code 7 control cards.

9. Supply the tables that you are using. Define the location (in your deck) of each of the table array numbers on code 8 control cards and then place all of the tables in that specified order behind a single code 9 control card.

FORTRAN LISTINGS OF EXECUTIVE ROUTINES

```
*DECK TR1
C       03/13/75    12.56.47     JOHN HOLMES                                    TR100010
        OVERLAY(TRAJ,0,0)                                                       TR100100
        PROGRAM OV (INPUT,OUTPUT,TAPE5,TAPE19,TAPE9,TAPE6=OUTPUT)
C       8/2/77     JOHN HOLMES
        COMMON  Y(4940)                                                         TR100140
        COMMON/TAB/Z(50)                                                        TR100150
        EQUIVALENCE(Y(2311),NOPLOT)                                             TR100160
  100   CALL ZERO                                                               TR100170
  200   CALL INPUT                                                              TR100180
  300   CALL OVERLAY(4HTRAJ,1,0,6HRECALL)
        IF(NOPLOT.GT.0) GO TO 900                                               TR100200
500     CALL RESET                                                              TR100210
        GO TO 300                                                               TR100220
  900   CALL OVERLAY(4HTRAJ,2,0,6HRECALL)
        GO TO 500                                                               TR100240
 1000   STOP                                                                    TR100250
         END                                                                    TR100260
                                                                                TR100270
                                                                                TR100280
        SUBROUTINE ZERO                                                         TR100290
C       7/2/74     JOHN E. HOLMES                                               TR100300
        COMMON  Y(4940)                                                         TR100310
        EQUIVALENCE (Y(2302),J),(Y(2305),MPR)                                   TR100320
        EQUIVALENCE (Y(2304),L),(Y(2301),XNE)                                   TR100330
        EQUIVALENCE (Y(2306),ERROR)                                             TR100340
        EQUIVALENCE(Y(3000),RE),(Y(3001),WE)                                    TR100350
        DO 1 I=1,4940                                                           TR100360
      1 Y(I)=0.0                                                                TR100370
C                                                                               TR100380
C       DEFAULT OPTIONS                                                         TR100390
C                                                                               TR100400
C        J=2                                                                    TR100410
        Y(2302)=2.0                                                             TR100420
C        MPR=1                                                                  TR100430
        Y(2305)=1.0                                                             TR100440
C        L=0                                                                    TR100450
        Y(2304)=0.0                                                             TR100460
        XNE=0.0                                                                 TR100470
        ERROR=-1.0                                                              TR100480
        WE=0.0000729211508                                                      TR100490
C                                                                               TR100500
C       MEAN RADIUS FOR SPHERICAL EARTH                                         TR100510
        RE=20925631.                                                            TR100520
C                                                                               TR100530
        RETURN                                                                  TR100540
         END                                                                    TR100550
                                                                                TR100560
                                                                                TR100570
        SUBROUTINE INPUT                                                        TR100580
C       7/2/74     JOHN E. HOLMES                                               TR100590
        COMMON  Y(4940)                                                         TR100600
        INTEGER OUTNO,STPNO(10),PLOT(10),CVAR(31),DVAR(31)                      TR100610
        EQUIVALENCE(Y(2307),NOMOD),(Y(2312),NMOD(1))                            TR100620
        EQUIVALENCE(Y(2308),NOOUT),(Y(2332),RNME1(1)),(Y(2392),OUTNO(1))        TR100630
        EQUIVALENCE(Y(2309),NOIN),(Y(2422),INNO(1)),(Y(2622),VALVE(1))          TR100650
        EQUIVALENCE(Y(2310),NOSTOP),(Y(2822),STPNO(1)),(Y(2832),SUP(1)),        TR100660
       +(Y(2842),SLO(1))                                                        TR100670
        EQUIVALENCE(Y(2311),NOPLOT),(Y(2852),PLOT(1))                           TR100680
        EQUIVALENCE(Y(2872),LOCC(1)),(Y(2903),CVAR(1)),(Y(2934),LOCD(1))        TR100690
        EQUIVALENCE(Y(2965),DVAR(1)),(Y(4890),KTAB(1)),(Y(2871),NOTAB)          TR100700
        EQUIVALENCE(Y(2870),NODER),(Y(2869),NOVAR)                              TR100710
```

```
      EQUIVALENCF(Y(3051),K(1)),(Y(3044),TITL(1))                       TR100720
      EQUIVALENCF(Y(2996),KEND)                                         TR100730
      EQUIVALENCF(Y(2362),HD1(1)),(Y(2377),HD2(1))                      TR100735
      DIMENSION TITL(7),NMOD(20),RNME1(30),OUTNO(30),INNO(200)          TR100740
     +,VALVE(200),SUP(10),SLO(10),K(49),                                TR100750
     +LOCC(31),LOCD(31),KTAB(49)                                        TR100760
      DIMENSION HD1(15),HD2(15)                                         TR100765
      NOIN=0                                                            TR100770
      NOMOD=0                                                           TR100780
      NOOUT=0                                                           TR100790
      NOSTOP=0                                                          TR100800
      NOPLOT=0                                                          TR100810
      NOVAR=0                                                           TR100820
      NODER=0                                                           TR100830
      NOTAB=0                                                           TR100840
      CK=10H                                                            TR100842
      DO 970 I=1,15                                                     TR100844
      HD1(I)=5HF8.0                                                     TR100845
  970 HD2(I)=5HF8.0                                                     TR100846
      DO 101 I=1,49                                                     TR100850
  101 K(I)=0                                                            TR100860
      READ(5,998) IR,(TITL(I),I=1,7)                                    TR100870
      XDATE=DATE(DUM)                                                   TR100872
      XTIME=TIME(DUMY)                                                  TR100874
      WRITE(6,5000)                                                     TR100876
 5000 FORMAT(1H1,T56,*NAVAL SURFACE WEAPONS CENTER*/                    TR100878
     +T59,*WHITE OAK LABORATORY*)                                       TR100879
      WRITE(6,5001) (TITL(I),I=1,7),XDATE,XTIME                         TR100880
 5001 FORMAT(1H0,10X,7A10,T87,*RUN DATE*,A10,T107,*TIME*,A10/)          TR100882
 2000 FORMAT(1H1)                                                       TR100890
  998 FORMAT(I2,7A10)                                                   TR100900
      IPRT=1                                                            TR100905
    1 READ(5,1000) IR,IN,VAR,VARR,HED1,HED2,HED3                        TR100910
 1000 FORMAT(I2,I6,2E14.6,A10,A7,A10)                                   TR100915
      GO TO(2,3) IPRT                                                   TR100920
    2 IPRT=2                                                            TR100925
      IR1=IR$IN1=IN$VAR1=VAR$VARR1=VARR$HED11=HED1$HED21=HED2$HED31=HED3TR100930
      GO TO 4                                                           TR100940
    3 IPRT=1                                                            TR100945
      WRITE(6,1001)IR1,IN1,VAR1,VARR1,HED11,HED21,HED31,IR,IN,VAR,      TR100950
     +VARR,HED1,HED2,HED3                                               TR100955
 1001 FORMAT(1H ,I2,I6,2E14.6,A10,A7,A10,3X,2H**,3X,I2,I6,              TR100960
     +2E14.6,A10,A7,A10)                                                TR100965
    4 CONTINUE                                                          TR100968
C                                                                       TR100970
C     MODULE NUMBER                                                     TR100980
C                                                                       TR100990
    5 IF(IR.NE.1) GO TO 10                                              TR101000
      NOMOD=NOMOD+1                                                     TR101010
      IF(NOMOD.GT.20) GO TO 6                                           TR101020
      GO TO 7                                                           TR101030
    6 WRITE(6,4001)                                                     TR101040
 4001 FORMAT(1H0,*THE NUMBER OF MODULES EXCEEDS 20*)                    TR101050
      STOP                                                              TR101060
    7 CONTINUE                                                          TR101070
      NMOD(NOMOD)=IN                                                    TR101080
      GO TO 1                                                           TR101090
C                                                                       TR101100
C     VARIABLE TO BE LISTED IN OUTPUT                                   TR101110
C                                                                       TR101120
   10 IF(IR.NE.2) GO TO 15                                              TR101130
      NOOUT=NOOUT+1                                                     TR101140
      IF(NOOUT.GT.30) GO TO 11                                          TR101150
      GO TO 12                                                          TR101160
   11 WRITE(6,4002)                                                     TR101170
 4002 FORMAT(1H0,*PRINTOUT OF MORE THAN 30 ITEMS WAS CONSIDERED         TR101180
```

```
     +EXCESSIVE AND THEY WERE DROPPED*)                                 TR101190
      NOOUT=NOOUT-1                                                     TR101195
      GO TO 1                                                           TR101200
12    CONTINUE                                                          TR101210
      RNME1(NOOUT)=HED1                                                 TR101220
      IF(HED3.EQ.CK) GO TO 230                                          TR101222
      IF(NOOUT.GT.15) GO TO 220                                         TR101225
      HD1(NOOUT)=HED3                                                   TR101227
      GO TO 230                                                         TR101229
220   NOT=NOOUT-15                                                      TR101231
      HD2(NOT)=HED3                                                     TR101233
230   CONTINUE                                                          TR101235
      OUTNO(NOOUT)=IN                                                   TR101240
      GO TO 1                                                           TR101250
C                                                                       TR101260
C     DATA LOCATION AND VALUE                                           TR101270
C                                                                       TR101280
   15 IF(IR.NE.3) GO TO 20                                              TR101290
      NOIN=NOIN+1                                                       TR101300
      IF(NOIN.GT.200) GO TO 16                                          TR101310
      GO TO 17                                                          TR101320
16    WRITE(6,4003)                                                     TR101330
4003  FORMAT(1H0,*NUMBER OF INPUT VARIABLES EXCEEDS 200*)               TR101340
      STOP                                                              TR101350
17    CONTINUE                                                          TR101360
      INNO(NOIN)=IN                                                     TR101370
      VALVE(NOIN)=VAR                                                   TR101380
      GO TO 1                                                           TR101390
C                                                                       TR101400
C     TERMINATION VARIABLE WITH UPPER AND LOWER BOUNDS                  TR101410
C                                                                       TR101420
   20 IF(IR.NE.4) GO TO 30                                              TR101430
      NOSTOP=NOSTOP+1                                                   TR101440
      IF(NOSTOP.GT.10) GO TO 21                                         TR101450
      GO TO 22                                                          TR101460
21    WRITE(6,4004)                                                     TR101470
4004  FORMAT(1H0,*NUMBER OF STOP CONDITIONS EXCEEDS 10*)                TR101480
      STOP                                                              TR101490
22    CONTINUE                                                          TR101500
      STPNO(NOSTOP)=IN                                                  TR101510
      SUP(NOSTOP)=VARR                                                  TR101520
      SLO(NOSTOP)=VAR                                                   TR101530
      GO TO 1                                                           TR101540
C                                                                       TR101550
C     VARIABLES TO BE PLOTTED                                           TR101560
C                                                                       TR101570
   30 IF(IR.NE.5) GO TO 40                                              TR101580
      NOPLOT=NOPLOT+1                                                   TR101590
      IF(NOPLOT.GT.10) GO TO 31                                         TR101600
      GO TO 32                                                          TR101610
31    WRITE(6,4005)                                                     TR101620
4005  FORMAT(1H0,*NUMBER OF PLOT VARIABLES EXCEEDS 10*)                 TR101630
      STOP                                                              TR101640
32    CONTINUE                                                          TR101650
      PLOT(NOPLOT)=IN                                                   TR101660
      GO TO 1                                                           TR101670
C                                                                       TR101680
C     DEPENDENT VARIABLES, C ARRAY                                      TR101690
C                                                                       TR101700
   40 IF(IR.NE.6) GO TO 50                                              TR101710
      NOVAR=NOVAR+1                                                     TR101720
      IF(NOVAR.GT.28) GO TO 42                                          TR101730
      GO TO 43                                                          TR101740
42    WRITE(6,4000)                                                     TR101750
4000  FORMAT(1H0,*THE NUMBER OF DIFFERENTIAL EQUATIONS EXCEEDS 28*)     TR101760
      STOP                                                              TR101770
```

```
43    CONTINUE
      LOCC(NOVAR)=VAR
      CVAR(NOVAR)=IN
      GO TO 1
C
C     DERIVATIVES,  D ARRAY
C
   50 IF(IR.NE.7) GO TO 60
      NODER=NODER+1
      LOCD(NODER)=VAR
      DVAR(NODER)=IN
      GO TO 1
C
C     TABULATED VALUES
   60 IF(IR.NE.8) GO TO 70
      NOTAB=NOTAB+1
      IF(NOTAB.GT.49) GO TO 61
      GO TO 62
61    WRITE(6,4006)
4006  FORMAT(1H0,*THE NUMBER OF TABLES EXCEEDS 49*)
      STOP
62    CONTINUE
      IVAR=VAR
      KTAB(IVAR)=IN
      GO TO 1
C
C     AN INDICATER THAT TABLES ARE TO BE READ
C
   70 IF(IR.NE.9) GO TO 80
      CALL  TAPTAB(K,KEND)
      GO TO 1
80    CONTINUE
C     END OF DATA FOR A SINGLE RUN
90    WRITE(6,2000)
      RETURN
       END





      SUBROUTINE  RESET
C     7/2/74     JOHN E. HOLMES
      COMMON   Y(4940)
      INTEGER STPNO(10)
      EQUIVALENCE(Y(2863),STOP)
      EQUIVALENCE(Y(2307),NOMOD),(Y(2312),NMOD(1))
      EQUIVALENCE(Y(2309),NOIN),(Y(2422),INNO(1)),(Y(2622),VALVE(1))
      EQUIVALENCE(Y(2871),NOTAB),(Y(4890),KTAB(1))
      EQUIVALENCE(Y(2306),ERROR),(Y(2996),STAGE)
      EQUIVALENCE(Y(3044),TITL(1)),(Y(2301),XNE)
      EQUIVALENCE(Y(2310),NOSTOP)
      EQUIVALENCE(Y(2822),STPNO(1)),(Y(2832),SUP(1)),(Y(2842),SLO(1))
      DIMENSION TITL(7),NMOD(20),INNO(200),VALVE(200),KTAB(49)
      DIMENSION SUP(10),SLO(10)
      READ(5,998) IR,(TITL(I),I=1,7)
C     IR = 99, STOP.ALL RUNS FINISHED
      IF(IR.EQ.99) STOP
C     IR = 11,STAGING HAS OCCURRED, NEW MODS AND TABLES ARE TO BE USED
      IF(IR.EQ.11) GO TO 80
      DO 10 I=1,2307
   10 Y(I)=0.0
      DO 12 I=2862,2868
   12 Y(I)=0.0
      Y(2998)=0.0
      Y(2999)=0.0
```

```
      REWIND 19
      DO 14 I=3002,3043
   14 Y(I)=0.0
C*******************
C     DEFAULT OPTIONS
C
C     J=2
      Y(2302)=2.0
C     MPR=1
      Y(2305)=1.0
C     L=0
      Y(2304)=0.0
      XNE=0.0
      ERROR=-1.0
C*******************
      GO TO 85
   80 CONTINUE
        STOP=0.
      STAGE=STAGE+1.0
85    CONTINUE
      WRITE(6,999) (TITL(I),I=1,7)
      NOMOD=0
      NOSTOP=0
      NOTAB=0
100   READ(5,1000) IR,IN,VAR,VARR,HED1,HED2,HED3,HED4
      WRITE(6,1001) IR,IN,VAR,VARR,HED1,HED2,HED3,HED4
1000  FORMAT(I2,I6,2E14.6,A10,A7,A10,A9)
1001  FORMAT(1X,I2,I6,2F14.6,A10,A7,A10,A9)
C     MODULE NUMBER
      IF(IR.NE.1) GO TO 110
      NOMOD=NOMOD+1
      NMOD(NOMOD)=IN
      GO TO 100
C     DATA LOCATION AND VALUE
  110 IF(IR.NE.3) GO TO 120
      NOIN=NOIN+1
      INNO(NOIN)=IN
      VALVE(NOIN)=VAR
      GO TO 100
120   IF(IR.NE.4) GO TO 125
C     TERMINATION VARIABLES
      NOSTOP=NOSTOP+1
      STPNO(NOSTOP)=IN
      SUP(NOSTOP)=VARR
      SLO(NOSTOP)=VAR
      GO TO 100
C     TABULATED VALUES
125   IF(IR.NE.8) GO TO 130
      NOTAB=NOTAB+1
      IVAR=VAR
      KTAB(IVAR)=IN
      GO TO 100
C     END OF DATA FOR A SINGLE RUN
  130 IF(IR.NE.10) GO TO 100
      WRITE(6,2000)
 2000 FORMAT(1H1)
  999 FORMAT(1H0,7A10)
  998 FORMAT(I2,7A10)
      RETURN
      END
      SUBROUTINE TAPTAB(K,KEND)
C
C     7/2/74     JOHN E. HOLMES
C
C     IT READS THE DATA CARDS FOR THE TABLES , ARRANGES THEM INTO THE
C     TABLE ARRAY, AND WRITES THE TABLE ARRAY ON FILE 9
```

```
C                                                                        TR103090
C     TAPTAB IS A SUBROUTINE TO BE USED WITH FRMRAN.                     TR103100
C                                                                        TR103110
C                                                                        TR103120
C         TABLE (1) = N(1)        NUMBER OF ARGUMENTS FOR FIRST VARIABLE TR103130
C         TABLE (2) = N(2)        NUMBER OF ARGUMENTS FOR SECOND VARIABLE TR103140
C         ...                                                            TR103150
C         TABLE (N) = N(N)        NUMBER OF ARGUMENTS FOR N-TH VARIABLE  TR103160
C         TABLE(N+1) = U(1,1)     FIRST OF ARGUMENTS CORRESPONDING TO    TR103170
C                                 THE FIRST VARIABLE                     TR103180
C         ...                                                            TR103190
C         TABLE(N+N(1)) = U(1,N(1))    LAST OF ARGUMENTS CORRESPONDING   TR103200
C                                 TO THE FIRST VARIABLE                  TR103210
C         TABLE(N+N(1)+1) = U(2,1)  FIRST OF THE ARGUMENTS CORRESPONDING TR103220
C                                 TO THE SECOND VARIABLE                 TR103230
C         ...                                                            TR103240
C         TABLE(N+N(1)+N(2)) = U(2,N(2))  LAST OF THE ARGUMENTS          TR103250
C                                 CORRESPONDING TO THE SECOND VARIABLE   TR103260
C         ...                                                            TR103270
C         TABLE(N+N(1)+N(2)+...+N(N)) = U(N,N(N))   LAST OF ARGUMENTS    TR103280
C                                 CORRESPONDING TO THE N TH VARIABLE     TR103290
C         TABLE(N+N(1)+N(2.+...+N(N)+1) = ARG(1,1,...,1)   TABLE VALUE   TR103300
C                                 CORRESPONDING TO THE FIRST ARGUMENT OF TR103310
C                                 EACH VARIABLE                          TR103320
C         TABLE(N+N(1)+N(2)+...+N(N)+1) = ARG(1,1,...,1,2)               TR103330
C         ...                                                            TR103340
C         ...                                                            TR103360
C         TABLE(N+N(1)+N(2)+...+N(N)+N(1)+1) = ARG(1,1,....2,1)          TR103350
C         TABLE(N+N(1)+N(2)+...+N(N)+N(1)*N(2)*...*N(N)) = ARG(N(1),N(2) TR103370
C                                                          ,...,N(N))    TR103380
C                                                                        TR103390
C         READS IN THE ADDITIONAL TABLE VALUES STARTING IN THE NEXT      TR103400
C         AVAILABLE LOCATION IN THE TABLE ARRAY AND USING THE SAME       TR103410
C         ARRANGEMENT AS ABOVE                                           TR103420
C                                                                        TR103430
C                                                                        TR103440
C     FOR ADDITIONAL TABLES, TAPTAB READS THEM IN AS ABOVE, STARTING     TR103450
C     IN THE NEXT AVAILABLE LOCATION IN THE TABLE ARRAY                  TR103460
C                                                                        TR103470
C                                                                        TR103480
C              L,NUM,MFNC,N(1),N(2),...,N(NUM)     (WITH FORMAT 14I5)    TR103530
C         THE CONTROL CARDS MUST BE OF THE FOLLOWING FORMAT              TR103520
C                                                                        TR103510
C         M(J) = NUMBER OF FUNCTIONS TABULATED FOR THE J-TH TABLE        TR103500
C         NI(J)= NUMBER OF VARIABLES IN THE J-TH TABLE                   TR103490
C         L=0,NUM NE 0    ONE TABLE OF NUM INDEPENDENT VARIABLES IS READ TR103540
C                             THE DECK OF THE TABLE IS AS FOLLOWS        TR103550
C                             N(1) ARGUMENTS (VALUES) OF THE FIRST       TR103560
C                             VARIABLE - IN ASCENDING ORDER WITH SIX PER TR103570
C                             CARD                                       TR103580
C                             N(2) ARGUMENTS OF THE SECOND VARIABLE,     TR103590
C                             BEGINNING ON A NEW CARD, IN ASCENDING ORDERTR103600
C                             WITH SIX PER CARD                          TR103610
C                             ...                                        TR103620
C                             N(NUM) ARGUMENTS OF U(NUM), BEGINNING ON A TR103630
C                             NEW CARD, IN ASCENDING ORDER WITH SIX PER  TR103640
C                             CARD.                                      TR103650
C                             THE FIRST FUNCTION TABULATED IS PUNCHED IN TR103660
C                             BLOCKS OF N(NUM) WITH SIX PER CARD.  THE   TR103670
C                             FIRST BLOCK STARTS WITH THE VALUE          TR103680
C                             CORRESPONDING TO THE SMALLEST ARGUMENT OF  TR103690
C                             EACH VARIABLE AND ALLOWS U(N) TO VARY.     TR103700
C                             THE NEXT BLOCK STARTS ON A NEW CARD AND HASTR103710
C                             U(N-1) INCREMENTED AND ALLOWS U(N) TO VARY TR103720
C                             AND SO ON UNTIL THE ENTIRE TABULATION IS   TR103730
C                             COMPLETED.                                 TR103740
```

```
C                              IF MFNC IS GREATER THAN ONE, REPEAT FOR
C                              EACH SUCCEEDING FUNCTION TO BE INCLUDED IN
C                              THE TABLE.
C
C         L=0,NUM=0   TERMINATE THE RUN
C
C
C
      DIMENSION K(1),TABLE(100),N(15),M(49),T(7),NI(49)
      EQUIVALENCE (NARG,ARG)
      KOUNT = 1
      NEXLOC = 0
      DO 310 I=1,100
  310 TABLE(I)=0.
300   READ(5,201)  (T(I),I=1,7),DUM
      WRITE(6,202)
      WRITE(6,201)   (T(I),I=1,7),DUM
      READ(5,100) L,NUM,M(KOUNT),(N(I),I=1,NUM),DUM
      WRITE(6,100) L,NUM,M(KOUNT),(N(I),I=1,NUM),DUM
      IF (L.NE.0.OR.NUM.GT.15.OR.M(KOUNT).GT.15)   GOTO 800
      IF (NUM.EQ.0) RETURN
      NI(KOUNT) = NUM
      K(KOUNT)=NEXLOC+1
C
C         TO CALCULATE THE PRODUCT OF THE N(I) S
C         AND PUT THE N(I) S  IN THE TABLE
C
      NSUM = 0
      NPROD = 1
      DO 400 I=1,NUM
      NARG = N(I)
      NSUM = NSUM + NARG
      NPROD = NPROD * NARG
      TABLE(I) = ARG
  400 CONTINUE
      BUFFER OUT (9,1)(TABLE(1),TABLE(NUM))
      IF(UNIT(9)) 450, 450,1000
450   CONTINUE
C
C         TO PUT U(I) S  IN THE TABLE
C
      KSTART = NUM + NEXLOC + 1
      DO 500 I=1,NUM
      KEND = KSTART + N(I)- 1
      KKEND=KEND-KSTART+1
      IF(KKEND.GE.99) GO TO 900
      READ(5,110)   (TABLE(KK),KK=1,KKEND)
      WRITE(6,203)  (TABLE(KK),KK=1,KKEND)
      BUFFER OUT (9,1)(TABLE(1),TABLE(KKEND))
      IF (UNIT(9)) 490,490,1000
490   CONTINUE
      KSTART = KEND + 1
  500 CONTINUE
C
C         TO PUT THE TABLUATED VALUES IN THE TABLE
C
      KADD = N(NUM) - 1
      IEND = NPROD * M(KOUNT) / N(NUM)
      DO 600 I=1,IEND
      KEND = KSTART + KADD
      KKEND=KEND-KSTART+1
      IF(KKEND.GE.99) GO TO 900
      READ(5,110)   (TABLE(KK),KK=1,KKEND)
      WRITE(6,203)  (TABLE(KK),KK=1,KKEND)
      BUFFER OUT (9,1)(TABLE(1),TABLE(KKEND))
      IF(UNIT(9)) 580,580,1000
```

```
580     CONTINUE                                                           TR104420
        KSTART = KEND + 1                                                  TR104430
  600   CONTINUE                                                           TR104440
        NEXLOC=KEND                                                        TR104450
        KOUNT = KOUNT + 1                                                  TR104460
        IF(KEND.GE.3000)GO TO 950                                          TR104470
        GO TO 300                                                          TR104480
  800   WRITE(6,200)                                                       TR104490
        STOP                                                               TR104500
  900   WRITE(6,220)                                                       TR104510
        STOP                                                               TR104520
  950   WRITE(6,230)                                                       TR104530
        STOP                                                               TR104540
1000    WRITE(6,240)                                                       TR104550
        STOP                                                               TR104560
100     FORMAT(14I5,2X,A8)                                                 TR104570
  110   FORMAT (6E12.5)                                                    TR104580
  200   FORMAT(1H0,* TAPTAB REQUIRES THAT L BE EQUAL TO 0 AND THAT N       TR104590
       +AND M BE BETWEEN 0 AND 15 --ALSO CHECK FOR INCORRECT NUMBER        TR104600
       +OF DATA CARDS  *)                                                  TR104610
201     FORMAT (1X,7A10,1X,A8)                                             TR104620
  202   FORMAT (1H0)                                                       TR104630
  220   FORMAT(1H0,*  TAPTAB IS ATTEMPTING TO READ MORE THAN 99 VALUES     TR104650
       1AT ONE TIME, THE DIMENSION OF THE TABLE ARRAY WILL HAVE TO
       2BE INCREASED*)
  203   FORMAT (1P6E20.6)                                                  TR104640
  230   FORMAT(1H0,*  TABLES EXCEED DIMENSIONED SIZE OF 3000 *)            TR104680
240     FORMAT(1H0,* PARITY ERROR OCCURRED DURING THE WRITING              TR104690
       +OF THE TABLES ONTO UNIT 9  *)                                      TR104700
        END
        OVERLAY(TRAJ,1,0)                                                  TR104720
        PROGRAM START
C       7/2/74     JOHN E. HOLMES                                          TR104740
        COMMON  Y(4940)                                                    TR104750
        COMMON/TABL/TABLE(3000)                                            TR104760
         EQUIVALENCE(Y(2309),NOIN),(Y(2996),STAGE)                         TR104770
        EQUIVALENCE(Y(2422),INNO(1)),(Y(2622),VALVE(1)),(Y(2996),KEND)     TR104780
        DIMENSION  INNO(200),VALVE(200)                                    TR104790
        REWIND 9                                                           TR104800
        DO 1001    I=1,NOIN                                                TR104810
        IN=INNO(I)                                                         TR104820
        VAR=VALVE(I)                                                       TR104830
        Y(IN)=VAR                                                          TR104840
 1001   CONTINUE                                                           TR104850
        IF(STAGE.GT.0.0) GO TO 1005                                        TR104860
        LLEN=0                                                             TR104870
        N=1                                                                TR104880
        M=3000                                                             TR104890
1003    BUFFER IN (9,1)(TABLE(N),TABLE(M))                                 TR104900
        IF(UNIT(9)) 10,12,11                                               TR104910
10      LEN=LENGTH(9)                                                      TR104920
        LLEN=LEN+LLEN                                                      TR104930
        IF(LLEN.EQ.3000) GO TO 12                                          TR104940
        N=LLEN+1                                                           TR104950
        GO TO 1003                                                         TR104960
11      WRITE(6,2000)                                                      TR104970
2000    FORMAT(1H0,* PARITY ERROR OCCURRED WHILE ATTEMPTING TO             TR104980
       +READ TABLES FROM UNIT 9 *)                                         TR104990
        STOP                                                               TR105000
12      CONTINUE                                                           TR105010
        CALL OVERLAY(4HTRAJ,1,1,6HRECALL)
 1005   CALL OVERLAY(4HTRAJ,1,2,6HRECALL)
        END                                                                TR105080
        SUBROUTINE SENCOS(A,SA,CA,KEY)                                     TR105090
        IF(KEY.NE.0) GO TO 10                                              TR105100
        AA=A/57.2957795                                                    TR105110
```

```
      SA=SIN(AA)
      CA=COS(AA)
      RETURN
 10   SA=SIN(A)
      CA=COS(A)
      RETURN
      END
      SUBROUTINE ARKTAN (A,B,C,MODE)
    3 IF (A) 10,4,10
    4 IF (B) 5,6,6
    5 Z=3.14159265
      GO TO 18
   6  Z=0.
      GO TO 21
  10  IF(B)13,11,17
  11  Z=SIGN(1.5707963,A)
  12  GO TO 18
  13  Z=ATAN(A/B)+SIGN(3.14159265,A)
  14  IF(Z-3.14159265)16,15,16
  15  Z=-Z
  16  GO TO 18
  17  Z=ATAN(A/B)
  18  IF(MODE)21,19,21
   19 C=57.2957795*Z
  20  GO TO 22
  21  C=Z
  22  IF(ABS(C)-1.E-07)23,23,24
  23  C=0.0
  24  RETURN
       END
      SUBROUTINE MATINV(A,B,C)
      DIMENSION A(9),B(9),C(9)
      DELA=A(1)*A(5)*A(9)+A(2)*A(6)*A(7)+A(3)*A(8)*A(4)-A(7)*A(5)*A(3)-A
     1(8)*A(6)*A(1)-A(9)*A(2)*A(4)
      IF(DELA)10,20,10
  10  J=0
      DO 11 I=1,7,3
      B(I)=A(J+1)
      B(I+1)=A(J+4)
      B(I+2)=A(J+7)
   11 J=J+1
      C(1)=(B(5)*B(9)-B(6)*B(8))/DELA
      C(2)=(B(6)*B(7)-B(4)*B(9))/DELA
      C(3)=(B(4)*B(8)-B(5)*B(7))/DELA
      C(4)=(B(3)*B(8)-B(2)*B(9))/DELA
      C(5)=(B(1)*B(9)-B(3)*B(7))/DELA
      C(6)=(B(2)*B(7)-B(1)*B(8))/DELA
      C(7)=(B(2)*B(6)-B(3)*B(5))/DELA
      C(8)=(B(3)*B(4)-B(1)*B(6))/DELA
      C(9)=(B(1)*B(5)-B(2)*B(4))/DELA
      RETURN
  20  WRITE (6,6)
   6  FORMAT(34H DETERMINANT OF A IS EQUAL TO ZERO)
      RETURN
       END
      SUBROUTINEARDCFT(H,PPZ,TTZ,RRZ,CCZ,GGZ)
      DIMENSION T(12),A(10)
C**** ATMOS COMPUTES STANDARD PRESSURE, TEMPERATURE, AND DENSITY RATIOS
C**** FOR A GIVEN ALTITUDE H IN FEET.
      X(Z,A,B,C)=A*ALOG(Z*(Z-B)+C)
      Y(Z,A,B,C)=A*ATAN(Z* B-C)
      DATA(A(I),I=1,10)/-3.350145769E-17,3.161762924E-14,-1.269919974E-1
     11,2.848535349E-9,-3.930824139E-7,3.432295909E-5,-1.832962145E-3,
     15.256403630E-2,-5.232974573E-1,-3.955242007/
      Z =0.0003048*H
      TM=0.
```

```
      DO 5 I=1,10
    5 TM=(TM+A(I))*Z
      TM= TM+283.7492391
      S1=Z+6356.77
      T( 1)=-1.4655396E- 7/S1
      T( 2)= 2.5653341E-11*ALOG(S1)
      T( 3)= 1.4116834E- 4*ALOG( 14.002385+Z)
      T( 4)=-3.8282910E- 5*ALOG(216.232250-Z)
      T( 5)=X(Z, 1.5084978E-4, 26.414270    ,  684.10967   )
      T( 6)=Y(Z, 6.7419880E-4,  0.044294588,    0.5850046 )
      T( 7)=X(Z, 8.5519675E-5,137.4745      ,10533.544     )
      T( 8)=Y(Z, 4.9863416E-5,  0.013120767,    0.90188546)
      T( 9)=X(Z,-2.5392354E-4,193.32352     ,10180.367     )
      T(10)=Y(Z, 1.1921879E-3,  0.034567717,    3.3413764 )
      T(11)=X(Z,-3.3888604E-5,384.32662     ,38131.516     )
      T(12)=Y(Z, 8.9812379E-5,  0.02881021 ,    5.5362654 )
      TINTEG=0.28016067E-02
      DO 10 I=1,12
   10 TINTEG=TINTEG+T(I)
      QINTEG=   -3483.6764*TINTEG
      TTZ=TM/288.16
      PPZ=EXP(QINTEG)
      RRZ=PPZ/TTZ
      CCZ=SQRT(401.874*TM)/340.294
      GGZ=(1./(1.+Z/6356.766))**2
      RETURN
      END
      SUBROUTINE FRMRAN (TABLE,NUM,MFNC,U,A)
      DIMENSION TABLE(1),U(1),A(1),T(1),TEMP(15),N(15),IDUM(1)
      COMMON /TAB/T
      EQUIVALENCE (N(1),TEMP(1)),(IDUM(1),T(1))
      NUMB = NUM
      NINS=2
      NSUM = 0
      NNPROD = 1
      DO 300 I=1,NUMB
      TEMP(I) = TABLE(I)
      NSUM = NSUM + N(I)
      NNPROD = NNPROD * N(I)
  300 CONTINUE
C
C       ************************************
C
C       TO FILL THE T ARRAY
C
      NUMT = 2 * NUMB
      JPOS = NUMB
      DO 400 I=1,NUMB
C
C       TO SPACE TO THE BEGINNING OF ARGUMENTS CORRESPONDING TO
C       THE I-TH VARIABLE
C
      JSTRT = JPOS + 2
      JPOS = JPOS + N(I)
      DO 410 J=JSTRT,JPOS
      IF(TABLE(J).GT.U(I)) GO TO 420
  410 CONTINUE
      J=JPOS
  420 IJ = NUMT + I
      IDUM(IJ) = J - JSTRT +2
C
C       T(1) THROUGH T(N) ARE THE ARGUMENTS CORRESPONDING TO THE N
C       VARIABLES
C
      T(I) = TABLE(J)
      IJ = I + NUMB
```

```
C
C     T(N+1) THROUGH T(2*N) ARE THE ARGUMENTS ONE POSITION BELOW THE
C     ABOVE ARGUMENTS
C
      T(IJ) = TABLE(J-1)
  400 CONTINUE
      ISTART = NUMT + 1
      NUMTB=NUMT+NUMB
      ISUM=IDUM(NUMTB)-1
      IF (NUMB.EQ.1) GO TO 440
      NINS=NUMB+1
      NTOP=NUMB-1
      NPROD = 1
      DO 430 I=1,NTOP
      IJ=NUMTB-I
      IIJ=NINS-I
      NPROD = NPROD * N(IIJ)
      ISUM = ISUM + (IDUM(IJ)-2) * NPROD
  430 CONTINUE
  440 IIDUM = NUMB + ISUM + NSUM
      DO 1000 M= 1,MENC
      IDUM(ISTART) = IIDUM
      INDEX = 1
      NPROD = 1
C
C     TO COMPUTE THE INDICES OF THE TABLE VALUES NEEDED FOR THE
C     INTERPOLATION
C
      DO 500 I=1,NUMB
      LNDEX = INDEX + NUMT
      DO 510 J=1,INDEX
      IJ = LNDEX + J
      KJ = NUMT + J
C
C     THE IDUM ARRAY CONTAINS THE VALUES OF THE INDICES OF THE
C     TABLE VALUES NEEDED FOR THE INTERPOLATION
C
      IDUM(IJ) = IDUM(KJ) + NPROD
  510 CONTINUE
      II=NINS-I
      NPROD = NPROD * N(II)
      INDEX = INDEX + INDEX
  500 CONTINUE
C
C     TO PUT THE TABLE VALUES NEEDED FOR THE INTERPOLATION IN THE
C     T ARRAY STARTING WITH T(2*N+1)
C
      DO 600 I=ISTART,IJ
      KJ = IDUM(I)
      T(I) = TABLE(KJ)
  600 CONTINUE
C
C     ***********************************
C
C     INTERPOLATION
C
      JEND = 2**NUMB + ISTART - 2
      KJ = NUMB + 1
      DO 700 I=1,NUMB
      IJ = KJ - I
      INDEX = NUMB + IJ
      TEM = (U(IJ)-T(INDEX))/(T(IJ)-T(INDEX))
      IJ = ISTART
      DO 710 J=ISTART,JEND,2
      T(IJ) = (T(J+1)-T(J))*TEM + T(J)
      IJ = IJ + 1
```

```
 710 CONTINUE                                                           TR107100
 700 CONTINUE                                                           TR107120
     JEND = (JEND+ISTART)/2                                             TR107110
     A(M) = T(NUMT+1)                                                   TR107130
C                                                                       TR107140
C        TO SPACE TO THE BEGINNING OF THE NEXT FUNCTION TABULATED       TR107150
C                                                                       TR107160
     IIDUM = IIDUM + NNPROD                                             TR107170
1000 CONTINUE                                                           TR107180
     RETURN                                                             TR107190
     END                                                                TR107200
     SUBROUTINE MATVEC(A,B,C,N)                                         TR107210
     DIMENSION A(9),B(9),C(9),F(9),G(9),H(9)                            TR107220
     IF (N) 10,6,10                                                     TR107230
  10 GO TO (5,6,5,6,5),N                                                TR107240
   6 DO 61 J=1,9                                                        TR107250
  61 F(J)=A(J)                                                          TR107260
     GO TO 70                                                           TR107270
   5 M2=1                                                               TR107280
     DO 36 K=1,3                                                        TR107290
     K1=K+6                                                             TR107300
     DO 36 J=K,K1,3                                                     TR107310
     F(M2)=A(J)                                                         TR107320
  36 M2=M2+1                                                            TR107330
  70 IF (N-1) 71,71,72                                                  TR107340
  71 M4=1                                                               TR107350
     DO 73 J=1,3                                                        TR107360
  73 G(J)=B(J)                                                          TR107370
     GO TO 80                                                           TR107380
  72 M4=7                                                               TR107390
     GO TO (74,74,74,75,75),N                                           TR107400
  74 DO 76 J=1,9                                                        TR107410
  76 G(J)=B(J)                                                          TR107420
     GO TO 80                                                           TR107430
  75 M2=1                                                               TR107440
     DO 66 K=1,3                                                        TR107450
     K1=K+6                                                             TR107460
     DO 66 J=K,K1,3                                                     TR107470
     G(M2)=B(J)                                                         TR107480
  66 M2=M2+1                                                            TR107490
  80 M2=1                                                               TR107500
     DO 30 M1=1,M4,3                                                    TR107510
     DO 30 K=1,3                                                        TR107520
     K1=K+6                                                             TR107530
     M3=M1                                                              TR107540
     H(M2)=0.                                                           TR107550
     DO 20 J=K,K1,3                                                     TR107560
     H(M2)=H(M2)+F(J)*G(M3)                                             TR107570
  20 M3=M3+1                                                            TR107580
  30 M2=M2+1                                                            TR107590
     IF (N-1) 95,95,96                                                  TR107600
  95 M1=3                                                               TR107610
     GO TO 90                                                           TR107620
  96 M1=9                                                               TR107630
  90 DO 91 J=1,M1                                                       TR107640
  91 C(J)=H(J)                                                          TR107650
     RETURN                                                             TR107660
      END                                                               TR107670
     SUBROUTINE ITAB(NTAB,N,U,V)                                        TR107680
C    7/2/74     JOHN E. HOLMES                                          TR107690
     COMMON  Y(4940)                                                    TR107700
     COMMON/TAB/Z(50)                                                   TR107710
     COMMON/TABI/TABLE(3000)                                            TR107720
     EQUIVALENCE(Y(4890),KTAB(1)),(Y(3051),K(1))                        TR107730
     DIMENSION KTAB(49),U(3),K(49)                                      TR107740
     IN=KTAB(NTAB)                                                      TR107750
```

```
      KK=K(IN)                                                          TR107755
      MFNC=1                                                            TR107760
      CALL FRMRAN(TABLE(KK),N,MFNC,U,V)                                 TR107800
      RETURN                                                            TR107810
      END                                                               TR107820
      SUBROUTINE FNOL3(J,NN,G,L,MPR,XNE,X,Y,D,DERIV,TERM,OUTPUT)        TR107830
      COMMON YY(4940)
      DIMENSION C(3),Y(31),YD(31),YP(31),YC(31),D(51),DM(31,4),DK(31,4) TR107840
     +,ERROR(31),YK(31)                                                 TR107850
      DATA EP6,EP11,M4/1.E-6,1.E-11,-4/                                 TR107860
      DATA (C(K),K=1,3)/2*.5,1./                                        TR107870
      DATA HMAX5/1.E35/                                                 TR107880
      NHTS=0                                                            TR107890
      EP2=0.                                                            TR107900
      N=NN                                                              TR107910
      JJ=J-2                                                            TR107920
      H=G                                                               TR107930
      HN=H                                                              TR107940
      MK=1                                                              TR107950
      NRET=M4                                                           TR107960
      JTEST=1                                                           TR107970
      IF (JJ .LT. 0) JTEST=4                                            TR107980
      IF (XNE.EQ. 0.) GO TO 15                                          TR107990
      EC=Y(N+3)                                                         TR108000
      EUP=10.**(-XNE)                                                   TR108010
      ELO=EUP*.001                                                      TR108020
      EM=ELO*31.6227766                                                 TR108030
   15 XD=X                                                              TR108040
      XS=XD                                                             TR108050
      DO 20 I=1,N                                                       TR108080
      ERROR(I)=0.                                                       TR108090
   20 YD(I)=Y(I)                                                        TR108100
      CALL DERIV(X,Y,D)                                                 TR108110
      CALL TERM (X,Y,D,F)                                               TR108120
C     PRINT                                                             TR108130
   50 CALL OUTPUT(X,Y,D,ERROR,N,L,H)                                    TR108140
***********************************************************************
*  NOTE  CHANGE MADE TO ORIGINAL FNOL3 FOR MODIFLY
      IF(YY(2863).EQ.1.0)  RETURN
***********************************************************************
      IF (NRET) 65,60,55                                                TR108150
   55 PRINT 3000, HN                                                    TR108160
 3000 FORMAT(108H1EXECUTION TERMINATED BECAUSE INTERVAL OF INTEGRATION LTR108170
     1ESS THAN 1.0E -6 TIMES INDEPENDENT VARIABLE (X).  H =,1PE15.7)    TR108180
      STOP                                                              TR108190
   60 RETURN                                                            TR108200
   65 NPR=0                                                             TR108210
      IF (MPR .LE. 0) PC=Y(N+1)                                         TR108220
  100 IF (JTEST .EQ. 5 .AND. H .NE. HN) GO TO 455                       TR108230
      IF (JJ .GE. 0) H=HN                                               TR108240
      IF (MK .NE. 0 .OR. JJ .NE. 0) GO TO 300                           TR108250
C------THE ADAMS MOULTON METHOD                                         TR108260
  200 HD24=H/24.                                                        TR108270
      JAM=0                                                             TR108280
      DO 210 I=1,N                                                      TR108290
      YPI=(55.*DM(I,1)+37.*DM(I,2))-(59.*DM(I,3)+9.*DM(I,4))            TR108300
      YP(I)=YD(I)+HD24*YPI                                              TR108310
      Y(I)=YP(I)                                                        TR108320
  210 CONTINUE                                                          TR108330
      X=XD+H                                                            TR108340
      CALL DERIV(X,Y,DM(1,4))                                           TR108350
      DO 220 I=1,N                                                      TR108360
      YPI=(9.*DM(I,4)+19.*DM(I,1)+DM(I,2))-5.*DM(I,3)                   TR108370
      YC(I)=YD(I)+HD24*YPI                                              TR108380
      ERROR(I)=(YP(I)-YC(I))/14.                                        TR108390
C THIS ADDS IN A 2D CORRECTION                                          TR108400
```

```
      YC(I)=YC(I)+ERROR(I)                                              TR108410
  220 CONTINUE                                                          TR108420
      IF (XNE.NE..0) GO TO 700                                          TR108430
      GO TO 455                                                         TR108440
C------THE RUNGE KUTTA METHOD                                            TR108450
300   GO TO (301,309,308,309,303),JTEST                                 TR108460
  301 DO 302 I=1,N                                                      TR108470
      YK(I)=YD(I)                                                       TR108480
  302 CONTINUE                                                          TR108490
      XDS=XD                                                            TR108500
      GO TO 309                                                         TR108510
  303 DO 304 I=1,N                                                      TR108520
      YK(I)=YC(I)                                                       TR108530
  304 CONTINUE                                                          TR108540
      XDS=XD+H                                                          TR108550
  308 HS=H                                                              TR108560
      H=2.*H                                                            TR108570
      GO TO 320                                                         TR108580
  309 X=XD                                                              TR108590
      JAM=1                                                             TR108600
      DO 310 I=1,N                                                      TR108610
      Y(I)=YD(I)                                                        TR108620
      DK(I,1)=D(I)                                                      TR108630
  310 CONTINUE                                                          TR108640
      IF (JTEST .LE. 2) CALL DERIV(X,Y,DK)                              TR108650
      IF (MK .GT. 1 .OR. JTEST .GT. 1) GO TO 320                        TR108660
      DO 315 I=1,N                                                      TR108670
      DM(I,4)=DK(I,1)                                                   TR108680
  315 CONTINUE                                                          TR108690
  320 DO 335 K=2,4                                                      TR108700
      HC=H*C(K-1)                                                       TR108710
      DO 330 I=1,N                                                      TR108720
      Y(I)=YD(I) + HC*DK(I,K-1)                                         TR108730
  330 CONTINUE                                                          TR108740
      X=XD+HC                                                           TR108750
      CALL DERIV(X,Y,DK(1,K))                                           TR108760
  335 CONTINUE                                                          TR108770
      HD6=H/6.                                                          TR108780
      DO 340 I=1,N                                                      TR108790
      YPI=DK(I,1)+DK(I,4)+2.*(DK(I,2)+DK(I,3))                          TR108800
      YC(I)=YD(I)+HD6*YPI                                               TR108810
  340 CONTINUE                                                          TR108820
      GO TO (360,390,370,455,370),JTEST                                 TR108830
  360 DO 365 I=1,N                                                      TR108910
      YP(I)=YC(I)                                                       TR108920
  365 CONTINUE                                                          TR108930
      JTEST=3                                                           TR108940
      GO TO 308                                                         TR108950
  370 DO 380 I=1,N                                                      TR108960
      YD(I)=YP(I)                                                       TR108970
      YP(I)=YC(I)                                                       TR108980
  380 CONTINUE                                                          TR108990
      H=HS                                                              TR109000
      XD=XD+H                                                           TR109010
      JTEST=2                                                           TR109020
      IF (MK .EQ. 1) GO TO 309                                          TR109030
      GO TO 451                                                         TR109040
  390 DO 400 I=1,N                                                      TR109050
      ERROR(I)=(YC(I)-YP(I))/15.                                        TR109060
      YC(I)=YC(I)+ERROR(I)                                              TR109070
      YP(I)=YC(I)                                                       TR109080
  400 CONTINUE                                                          TR109090
      JTEST=5                                                           TR109100
      IF (XNE.NE..0) GO TO 700                                          TR109110
C------ACCEPT THE STEP SIZE                                              TR109120
  450 IF (JAM .EQ. 0) GO TO 455                                         TR109130
```

```
      IF (MK .EQ. 3 .AND. JJ .EQ. 0) GO TO 455                          TR109140
      IF (MK .NE. 1) GO TO 303                                          TR109150
      IF (JTEST .EQ. 1) GO TO 455                                       TR109160
  451 DO 452 I=1,N                                                      TR109170
      Y(I)=YD(I)                                                        TR109180
  452 CONTINUE                                                          TR109190
      GO TO 466                                                         TR109200
  455 DO 459 NQ=1,N                                                     TR109210
      YD(NQ)=YC(NQ)                                                     TR109220
      Y(NQ)=YD(NQ)                                                      TR109230
  459 CONTINUE                                                          TR109240
      IF (JJ .GE. 0) JTEST=1                                            TR109250
      IF (MK .NE. 0 .OR. JJ .NE. 0 .OR. NRET .NE. M4) GO TO 465         TR109260
      DO 460 I=1,N                                                      TR109270
      DM(I,4)=DM(I,2)                                                   TR109280
      DM(I,2)=DM(I,3)                                                   TR109290
      DM(I,3)=DM(I,1)                                                   TR109300
  460 CONTINUE                                                          TR109310
  465 XD=XD+H                                                           TR109320
  466 X=XD                                                              TR109330
      IF (MK .EQ. 3) MK=0                                               TR109340
      CALL DERIV(X,Y,D)                                                 TR109350
      DO 470 I=1,N                                                      TR109360
      DM(I,MK+1)=D(I)                                                   TR109370
  470 CONTINUE                                                          TR109380
  480 FP=F                                                              TR109390
      CALL TERM (X,Y,D,F)                                               TR109400
C------DO YOU TERMINATE                                                 TR109410
  500 IF (ABS(F) .LE. EP6) GO TO 800                                    TR109420
      IF (FP .EQ. 0.) GO TO 510                                         TR109430
      IF (NRET .NE. M4 .OR. F*FP .LT. EP11) GO TO 805                   TR109440
  510 XS=XD                                                             TR109450
      IF (MK .NE. 0 .AND. H .EQ. HN) MK=MK+1                            TR109460
C------DO YOU PRINT                                                     TR109470
600   NPR=NPR+1                                                         TR109480
      IF (MPR .EQ. 0) GO TO 610                                         TR109490
      IF (NPR .GE. MPR) GO TO 50                                        TR109750
      GO TO 100                                                         TR109760
  610 IF (ABS(Y(N+1)-PC) .GE. Y(N+2)) GO TO 50                          TR109770
      GO TO 100                                                         TR109780
C------DETERMINING THE STEP SIZE                                        TR109790
  700 HB = HMAX5                                                        TR109800
      DO 760 I = 1,N                                                    TR109810
      Z=ABS(ERROR(I))                                                   TR10 820
      IF(YC(I).EQ.0.) GO TO 720                                         TR109825
      ZZ=YC(I)                                                          TR109830
      ZZ=ABS(ZZ)                                                        TR109840
      IF (EC) 720,710,705                                               TR109850
  705 IF (EC .GT. ZZ) ZZ=EC                                             TR109860
710   Z=Z/ZZ                                                            TR109870
  720 IF(Z.GT.ELO.AND.Z.LT.EUP) GOTO 750                                TR109880
      HB = AMIN1(HB,EM/(Z+EP11))                                        TR109890
      GOTO760                                                           TR109900
  750 HB=AMIN1(HB,1.)
  760 CONTINUE                                                          TR109920
      IF(HB.NE.1.) GO TO 765                                            TR110130
      NHTS=0                                                            TR110140
      GO TO 450                                                         TR110150
  765 HN=H*HB**.2                                                       TR110160
      IF (MK .NE. 1) JTEST=1                                            TR110180
      MK=1                                                              TR110190
      IF (HB.LT.1.) GOTO 770                                            TR110200
      IF (ABS(HN) .GT. ABS(4.*H)) HN=4.*H                               TR110210
      NHTS=0                                                            TR110220
      GOTO 450                                                          TR110230
  770 HEPS=ABS(X*EP6) + EP11                                            TR110240
```

```
      IF (ABS(HN) .LT. ABS(H/4.)) HN=H/4.                               TR110250
      IF (ABS(HN) .GT. HEPS) GO TO 790                                  TR110260
      NHTS = NHTS + 1                                                   TR110270
      IF (NHTS .LE. 10) GO TO 780                                       TR110280
      NRET = 1                                                          TR110290
      GO TO 450                                                         TR110300
  780 HN=SIGN(HEPS,HN)                                                  TR110310
      IF (NHTS .GT. 1) GO TO 450                                        TR110320
  790 IF (NHTS .GT. 1) NHTS=0                                           TR110330
      IF (JAM .EQ. 0) GO TO 100                                         TR110340
      DO 795 I=1,N                                                      TR110350
      YD(I)=YK(I)                                                       TR110360
  795 CONTINUE                                                          TR110370
      XD=XDS                                                            TR110380
      JTEST=1                                                           TR110390
      GO TO 100                                                         TR110400
C------THE TERMINATION LOOP                                             TR110410
  800 NRET=0                                                            TR110420
  805 IF (NRET .LT. 0) GO TO 806                                        TR110430
      H=XD-XS                                                           TR110440
      GO TO 50                                                          TR110450
  806 IF(F*FP.LT.0.) GOTO 810                                           TR110460
      IF (F*FP2.LT.0.) GOTO 820                                         TR110470
      GO TO 800                                                         TR110480
  810 FP2 =FP                                                           TR110490
      HP =H                                                             TR110500
      GOTO 830                                                          TR110510
  820 FP =FP2                                                           TR110520
      HP =H + HP                                                        TR110530
  830 NRET=NRET+1                                                       TR110540
      H=HP*F/(FP-F)                                                     TR110550
      JTEST=4                                                           TR110560
      GOTO 300                                                          TR110570
       END                                                              TR110580
      REAL FUNCTION KLMT(EI,L,K)
C
C     THIS IS USED FOR MODELING A RATE GYRO OR ANY OTHER PROPORTIONAL.
C              LIMITED OUTPUT DEVICE
C
C              EI = THE INPUT
C              L = THE LIMIT ON THE OUTPUT
C              K = THE GAIN OF THE DEVICE
C
      REAL L,K
      KLMT=K*EI
      IF(ABS(KLMT).GE.L) KLMT=SIGN(L,KLMT)
      END
```

```
*DECK TR2
C **********                                                            TR200010
C       03/13/75    12.56.47     JOHN HOLMES                            TR200012
C **********                                                            TR200014
      OVERLAY(TRAJ,1,1)                                                 TR200100
      PROGRAM SETUP                                                     TR200110
C     8/2/77     JOHN E. HOLMES
      COMMON  Y(4940)                                                   TR200130
      EQUIVALENCE (Y(2307),NOMOD),(Y(2312),NMOD(1))                     TR200140
      DIMENSION NMOD(20)                                                TR200150
      DO 1000  I=1,NOMOD                                                TR200160
      L=NMOD(I)                                                         TR200170
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),L      TR200180
    1 CALL IMOD1                                                        TR200190
      GO TO 1000                                                        TR200200
    2 CALL IMOD2                                                        TR200210
      GO TO 1000                                                        TR200220
    3 CALL IMOD3                                                        TR200230
      GO TO 1000                                                        TR200240
    4 CALL IMOD4                                                        TR200250
      GO TO 1000                                                        TR200260
    5 CALL IMOD5                                                        TR200270
      GO TO 1000                                                        TR200280
    6 CALL IMOD6                                                        TR200290
      GO TO 1000                                                        TR200300
    7 CALL IMOD7                                                        TR200310
      GO TO 1000                                                        TR200320
    8 CALL IMOD8                                                        TR200330
      GO TO 1000                                                        TR200340
    9 CALL IMOD9                                                        TR200350
      GO TO 1000                                                        TR200360
   10 CALL IMOD10                                                       TR200370
      GO TO 1000                                                        TR200380
   11 CALL IMOD11                                                       TR200390
      GO TO 1000                                                        TR200400
   12 CALL IMOD12                                                       TR200410
      GO TO 1000                                                        TR200420
   13 CALL IMOD13                                                       TR200430
      GO TO 1000                                                        TR200440
   14 CALL IMOD14                                                       TR200450
      GO TO 1000                                                        TR200460
   15 CALL IMOD15                                                       TR200470
      GO TO 1000                                                        TR200480
   16 CALL IMOD16                                                       TR200490
      GO TO 1000                                                        TR200500
   17 CALL IMOD17                                                       TR200510
      GO TO 1000                                                        TR200520
   18 CALL IMOD18                                                       TR200530
      GO TO 1000                                                        TR200540
   19 CALL IMOD19                                                       TR200550
      GO TO 1000                                                        TR200560
   20 CALL IMOD20                                                       TR200570
 1000 CONTINUE                                                          TR200580
       END                                                              TR200590
                                                                        TR200600
                                                                        TR200610
      SUBROUTINE IMOD1                                                  TR200620
      RETURN                                                            TR200630
      END                                                               TR200640
                                                                        TR200650
                                                                        TR200660
      SUBROUTINE IMOD2                                                  TR200670
```

```
      RETURN
      END

      SUBROUTINE IMOD3
      RETURN
      END

      SUBROUTINE IMOD4
      RETURN
      END

      SUBROUTINE IMOD5
      RETURN
      END

      SUBROUTINE IMOD6
      RETURN
      END

      SUBROUTINE IMOD7
      RETURN
      END

      SUBROUTINE IMOD8
      RETURN
      END

      SUBROUTINE IMOD9
      RETURN
      END

      SUBROUTINE IMOD10
      RETURN
      END

      SUBROUTINE IMOD11
      RETURN
      END

      SUBROUTINE IMOD12
      RETURN
      END

      SUBROUTINE IMOD13
      RETURN
      END

      SUBROUTINE IMOD14
      RETURN
      END

      SUBROUTINE IMOD15
      RETURN
```

```
      END

      SUBROUTINE IMOD16
      RETURN
      END

      SUBROUTINE IMOD17
      RETURN
      END

      SUBROUTINE IMOD18
      RETURN
      END

      SUBROUTINE IMOD19
      RETURN
      END

      SUBROUTINE IMOD20
      RETURN
      END

      OVERLAY(TRAJ,1,2)
      PROGRAM INTGRT
C     7/2/74     JOHN E. HOLMES
      COMMON  Y(4940)
      INTEGER CVAR(31),DVAR(31)
      EQUIVALENCE(Y(2300),G),(Y(2301),XNE)
      EQUIVALENCE(Y(2999),T)
      EQUIVALENCE(Y(2306),ERROR),(Y(2903),CVAR(1)),(Y(2934),LOCD(1))
      EQUIVALENCE(Y(2965),DVAR(1)),(Y(2872),LOCC(1)),(Y(2870),NODER)
      EQUIVALENCE(Y(2869),NOVAR)
      EQUIVALENCE(Y(2862),TI),(Y(2996),STAGE)
      EQUIVALENCE(Y(2868),DELT)
      DIMENSION LOCC(31),LOCD(31)
      DIMENSION C(31),D(31)
      EXTERNAL DERIV,TERM,OUT
      DO 10 I=1,31
      C(I)=0.0
10    D(I)=0.0
      DO 2000 I=1,NOVAR
      JJ=LOCC(I)
      JK=CVAR(I)
      C(JJ)=Y(JK)
 2000 CONTINUE
      DO 3000 I=1,NODER
      JJ=LOCD(I)
      JK=DVAR(I)
      D(JJ)=Y(JK)
 3000 CONTINUE
      IF(STAGE.GT.0.0)    TI=T
      T=TI
      C(NOVAR+2)=Y(2997)
      C(NOVAR+3)=ERROR
      NN=NOVAR
      DELT=G
      J=IFIX(Y(2302))
      L=IFIX(Y(2304))
      MPR=IFIX(Y(2305))
  400 CALL ENOL3(J,NN,G,L,MPR,XNE,T,C,D,DERIV,TERM,OUT)
```

```
      END                                                               TR202010
                                                                        TR202020
                                                                        TR202030
      SUBROUTINE DERIV(T,C,D)                                           TR202040
C     7/2/74      JOHN E. HOLMES                                        TR202050
      COMMON  Y(4940)                                                   TR202060
      INTEGER CVAR(31),DVAR(31)                                         TR202070
      EQUIVALENCE(Y(2307),NOMOD),(Y(2312),NMOD(1))                      TR202080
      EQUIVALENCE(Y(2903),CVAR(1)),(Y(2934),LOCD(1))                    TR202090
      EQUIVALENCE(Y(2965),DVAR(1)),(Y(2872),LOCC(1)),(Y(2870),NODER)    TR202100
      EQUIVALENCE(Y(2869),NOVAR)                                        TR202110
      DIMENSION LOCC(31),LOCD(31)                                       TR202120
      DIMENSION C(31),D(31),NMOD(20)                                    TR202130
      DO 2000 I=1,NOVAR                                                 TR202140
      JJ=LOCC(I)                                                        TR202150
      JK=CVAR(I)                                                        TR202160
      Y(JK)=C(JJ)                                                       TR202170
 2000 CONTINUE                                                          TR202180
      DO 1000 I=1,NOMOD                                                 TR202190
      L=NMOD(I)                                                         TR202200
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),L      TR202210
1     CALL MOD1                                                         TR202220
      GO TO 1000                                                        TR202230
2     CALL MOD2                                                         TR202240
      GO TO 1000                                                        TR202250
3     CALL MOD3                                                         TR202260
      GO TO 1000                                                        TR202270
4     CALL MOD4                                                         TR202280
      GO TO 1000                                                        TR202290
5     CALL MOD5                                                         TR202300
      GO TO 1000                                                        TR202310
6     CALL MOD6                                                         TR202320
      GO TO 1000                                                        TR202330
7     CALL MOD7                                                         TR202340
      GO TO 1000                                                        TR202350
8     CALL MOD8                                                         TR202360
      GO TO 1000                                                        TR202370
9     CALL MOD9                                                         TR202380
      GO TO 1000                                                        TR202390
10    CALL MOD10                                                        TR202400
      GO TO 1000                                                        TR202410
11    CALL MOD11                                                        TR202420
      GO TO 1000                                                        TR202430
12    CALL MOD12                                                        TR202440
      GO TO 1000                                                        TR202450
13    CALL MOD13                                                        TR202460
      GO TO 1000                                                        TR202470
14    CALL MOD14                                                        TR202480
      GO TO 1000                                                        TR202490
15    CALL MOD15                                                        TR202500
      GO TO 1000                                                        TR202510
16    CALL MOD16                                                        TR202520
      GO TO 1000                                                        TR202530
17    CALL MOD17                                                        TR202540
      GO TO 1000                                                        TR202550
18    CALL MOD18                                                        TR202560
      GO TO 1000                                                        TR202570
19    CALL MOD19                                                        TR202580
      GO TO 1000                                                        TR202590
20    CALL MOD20                                                        TR202600
 1000 CONTINUE                                                          TR202610
      DO 4500 I=1,NODER                                                 TR202620
      JJ=LOCD(I)                                                        TR202630
      JK=DVAR(I)                                                        TR202640
      D(JJ)=Y(JK)                                                       TR202650
 4500 CONTINUE                                                          TR202660
```

```
      RETURN                                                            TR202670
       END                                                              TR202680
                                                                        TR202690
                                                                        TR202700
      SUBROUTINE OUT(T,C,D,ERROR,DMY,DNYX,DELT)                         TR202710
C     7/2/74      JOHN E. HOLMES                                        TR202720
      COMMON  Y(4940)                                                   TR202730
      INTEGER CVAR(31),DVAR(31),PLOT(10),OUTNO(30)                      TR202740
      INTEGER PLCT                                                      TR202745
      EQUIVALENCE(Y(2862),TI),(Y(2863),STOP),(Y(2864),TS)               TR202750
      EQUIVALENCE(Y(2866),RUNN),(Y(2867),DATE)
      EQUIVALENCE(Y(2308),NOOUT),(Y(2332),NA1(1)),                      TR202770
     +(Y(2392),OUTNO(1))                                                TR202780
      EQUIVALENCE(Y(2311),NOPLOT),(Y(2852),PLOT(1))                     TR202790
      EQUIVALENCE(Y(2903),CVAR(1)),(Y(2934),LOCD(1))                    TR202800
      EQUIVALENCE(Y(2965),DVAR(1)),(Y(2872),LOCC(1)),(Y(2870),NODER)    TR202810
      EQUIVALENCE(Y(3044),TITL(1))                                      TR202820
      EQUIVALENCE(Y(2869),NOVAR)                                        TR202830
      EQUIVALENCE(Y(3043),PLCT),(Y(2362),HD1(1)),(Y(2377),HD2(1))       TR202835
      DIMENSION LOCC(31),LOCD(31)                                       TR202840
      DIMENSION TITL(7),HD1(15),HD2(15),FOR1(31),FOR2(31)               TR202850
      DIMENSION NA1(30),C(31),D(31)                                     TR202860
      REAL PL(10)                                                       TR202870
1     FORMAT(3X,4HRUN ,F12.2,10X,7A10,6X,7HFORMAT ,I1)                  TR202880
2     FORMAT(3X,F12.2,92X,5HPAGE ,I3)                                   TR202890
    4 FORMAT(1H1)                                                       TR202910
5     FORMAT(1H0,2X,5HTIME ,15A8/)                                      TR202920
** KZ = NO. LINES STORED FOR CURRENT PAGE
** NUM = NO. FORMATS
** NP = NO. PAGES OF EACH FORMAT
** J,J2 = SUBSCRIPTS FOR OUTPUT ARRAYS
      DO 3000 I=1,NODER                                                 TR202940
      JJ=LOCD(I)                                                        TR202950
      JK=DVAR(I)                                                        TR202960
      Y(JK)=D(JJ)                                                       TR202970
3000  CONTINUE                                                          TR202980
      DO 2000 I=1,NOVAR                                                 TR202990
      JJ=LOCC(I)                                                        TR203000
      JK=CVAR(I)                                                        TR203010
      Y(JK)=C(JJ)                                                       TR203020
 2000 CONTINUE                                                          TR203030
      Y(2868)=DELT                                                      TR203040
      Y(2999)=T                                                         TR203060
      IF(T-TI) 105,100,105                                              TR203080
  100 CONTINUE
      NP=0                                                              TR203100
      KZ=0                                                              TR203110
      J2=0
      J=0                                                               TR203120
  105 CALL PROCESS
*** STORE OUTPUT IN Y(3100+) ARRAY UNTIL FULL PAGE IS ACCUMULATED
  115 KZ=KZ+1
  200 J=J+1
      J2=J2+1
      Y(J+3100)=T
      Y(J2+3980)=T
      DO 330 I=1,NOOUT
      IN=OUTNO(I)
      IF(I.LE.15) J=J+1
      IF(I.LE.15) Y(J+3100)=Y(IN)
      IF(I.GT.15) J2=J2+1
      IF(I.GT.15) Y(J2+3980)=Y(IN)
  330 CONTINUE
  599 IF(NOPLOT.EQ.0) GO TO 331
      DO 600 I=1,NOPLOT
      II=PLOT(I)
```

```
  600 PL(I)=Y(II)
      PLCT=PLCT+1
      WRITE(19) RUNN,T,(PL(I),I=1,NOPLOT)
  331 IF(STOP) 332,332,15
  332 IF(KZ.LT.51) GO TO 333
*** WRITE PAGE OF OUTPUT IN EACH FORMAT..ARRAYS DIMENSIONED FOR 55 LINES
15    NP=NP+1                                                           TR203230
      IF(NP.GT.1) GO TO 22
      NUM=(14+NOOUT)/15                                                 TR203240
      FOR1(1)=9H(1X,F7.2,                                               TR203242
      FOR2(1)=9H(1X,F7.2,                                               TR203243
      DO 19 N=1,15
      I=2*N
      FOR1(I)=HD1(N)
      FOR1(I+1)=1H,
      IF(NOOUT.GE.N) GO TO 18
      FOR1(I)=10H
      FOR1(I+1)=1H
   18 FOR2(I)=HD2(N)
      FOR2(I+1)=1H,
      IF((NOOUT-15).GE.N) GO TO 19
      FOR2(I)=10H
      FOR2(I+1)=1H
   19 CONTINUE
      IF(NOOUT.GE.15) FOR1(31)=1H)
      IF(NOOUT.LT.15) FOR1(NOOUT*2+1)=1H)
      IF(NOOUT.LE.15) FOR2(1)=9H(1X)
      IF(NOOUT.GT.15) FOR2((NOOUT-15)*2+1)=1H)
   22 DO 30 L=1,NUM
      WRITE(6,1) RUNN,(TITL(I),I=1,7),L                                 TR203260
      WRITE(6,2) DATE,NP
      IF(L.GT.1) GO TO 21
20    WRITE(6,5) (NA1(I),I=1,15)                                        TR203290
      M=J+3100
      WRITE(6,FOR1) (Y(I),I=3101,M)                                     TR203320
      GO TO 26                                                          TR203330
21    WRITE(6,5) (NA1(I),I=16,30)                                       TR203340
      M=J2+3980
      WRITE(6,FOR2) (Y(I),I=3981,M)                                     TR203370
   26 WRITE(6,4)                                                        TR203530
   30 CONTINUE                                                          TR203540
      J=0                                                               TR203550
      J2=0
      KZ=0                                                              TR203580
      GO TO 333                                                         TR203590
  333 RETURN
       END                                                              TR204100
                                                                        TR204110
                                                                        TR204120
      SUBROUTINE TERM(T,C,D,F)                                          TR204130
C     7/2/74      JOHN E. HOLMES                                        TR204140
      COMMON  Y(4940)                                                   TR204150
      INTEGER CVAR(31),DVAR(31),STPNO(10)                               TR204160
      EQUIVALENCE  (Y(2862),TI)                                         TR204170
      EQUIVALENCE  (Y(2863),STOP)                                       TR204180
      EQUIVALENCE(Y(2310),NOSTOP),(Y(2822),STPNO(1)),(Y(2832),SUP(1)),  TR204190
     +(Y(2842),SLO(1))                                                  TR204200
      EQUIVALENCE(Y(2903),CVAR(1)),(Y(2934),LOCD(1)),(Y(2869),NOVAR)    TR204210
      EQUIVALENCE(Y(2965),DVAR(1)),(Y(2872),LOCC(1)),(Y(2870),NODER)    TR204220
      EQUIVALENCE(Y(2009),LRP(1))                                       TR204230
      DIMENSION LOCC(31),LOCD(31)                                       TR204240
      DIMENSION SUP(10),SLO(10),C(31),D(31),R(9),RPT(9),RPI(9)          TR204250
      REAL LRP(9)                                                       TR204260
      Y(2999)=T                                                         TR204270
7     F=1.0                                                             TR204280
      DELA=LRP(1)*LRP(5)*LRP(9)+LRP(2)*LRP(6)*LRP(7)+LRP(3)*            TR204290
```

```
      +LRP(8)*LRP(4)-LRP(7)*LRP(5)*LRP(3)-LRP(8)*LRP(6)*LRP(1)-
      +LRP(9)*LRP(2)*LRP(4)
       IF(DELA) 50,60,50
50     CONTINUE
C****TO INSURE ORTHOGONALITY OF LRP MATRIX    *****
       DO 9 I=1,3
       CALL MATINV(LRP,RPT,B)
       CALL MATINV(RPT,B,RPI)
       DO 8 J=1,9
8      LRP(J)=(LRP(J)+RPI(J))/2.
9      CONTINUE
60     CONTINUE
       DO 15 I=1,NODER
       JJ=LOCD(I)
       JK=DVAR(I)
       Y(JK)=D(JJ)
15     CONTINUE
       DO 10 I=1,NOVAR
       JJ=LOCC(I)
       JK=CVAR(I)
       Y(JK)=C(JJ)
   10  CONTINUE
       IPRVA=IFIX(Y(2998))
       C(NOVAR+1)=Y(IPRVA)
       SGN=1.
       DO 20  I=1,NOSTOP
       IN=STPNO(I)
       HALF=ABS(SUP(I)-SLO(I))/2.
       SMID=(SUP(I)+SLO(I))/2.
       FF=HALF-ABS(Y(IN)-SMID)
       IF(FF.LT.0.) SGN=-1.
       IF(FF.LE.0.) STOP=1.
       F=F*ABS(FF)
   20  CONTINUE
       F=F*SGN
   32  RETURN
        END

       SUBROUTINE MOD1
       RETURN
       END

       SUBROUTINE MOD2
       RETURN
       END

       SUBROUTINE MOD3
       RETURN
       END

       SUBROUTINE MOD4
       RETURN
       END

       SUBROUTINE MOD5
       RETURN
       END

       SUBROUTINE MOD6
       RETURN
```

```
      END


      SUBROUTINE MOD7
      RETURN
      END


      SUBROUTINE MOD8
      RETURN
      END


      SUBROUTINE MOD9
      RETURN
      END


      SUBROUTINE MOD10
      RETURN
      END


      SUBROUTINE MOD11
      RETURN
      END


      SUBROUTINE MOD12
      RETURN
      END


      SUBROUTINE MOD13
      RETURN
      END


      SUBROUTINE MOD14
      RETURN
      END


      SUBROUTINE MOD15
      RETURN
      END


      SUBROUTINE MOD16
      RETURN
      END


      SUBROUTINE MOD17
      RETURN
      END


      SUBROUTINE MOD18
      RETURN
      END


      SUBROUTINE MOD19
      RETURN
      END
```

```
      SUBROUTINE MOD20
      RETURN
      END


      SUBROUTINE PROCESS
      RETURN
      END

      OVERLAY(TRAJ,2,0)
      PROGRAM TRJPLTS
C     7/2/74     JOHN E. HOLMES
      COMMON  Y(4940)
      CALL MYPLOT
       END
      SUBROUTINE MYPLOT
C     11/13/74    JOHN E. HOLMES
      COMMON  Y(4940)
      INTEGER PLOT(10),PLCT
      EQUIVALENCE(Y(2311),NOPLOT),(Y(2852),PLOT(1))
      EQUIVALENCE(Y(3043),PLCT)
      REAL PL(10)
      REWIND 19
C
C     THE INFORMATION TO BE PLOTTED SHOULD BE READ AS FOLLOWS.
C
C     DO 20 J=1,PLCT
C     READ(19)   RUNN,T,(PL(I),I=1,NOPLOT)
C  20 CONTINUE
C  20 CONTINUE
C     PLCT ARE THE NUMBER OF DATA POINTS TO BE PLOTTED
C     SUPPLY YOUR OWN GOULD CALLS
C
      RETURN
       END
```

APPENDIX B

FIXED STORAGE ASSIGNMENTS

Y ARRAY LOCATIONS 2300 - 4940

NOTE:

An asterisk after the Y array location of the parameter indicates that the parameter is to be placed in the data as an input parameter.

PARAMETER	IDENTIFICATION	LOCATION IN Y ARRAY	
G		2300	*
XNE	INTEGRATION	2301	*
J	CONTROLS	2302	*
NN	SEE PAGE	2303	*
L		2304	
MPR		2305	*
ERROR		2306	*
NOMOD	Number of modules	2307	
NOOUT	Number of output parameters	2308	
NOIN	Number of code 3 parameters	2309	
NOSTOP	Number of STOP conditions	2310	
NOPLOT	Number of plot variables	2311	
NMOD(1)		2312	
↓	MOD controls, see subroutine INPUT	↓	
NMOD(20)		2331	
RNME1(1)		2332	
↓	Header controls, see subroutine INPUT	↓	
RNME1(30)		2361	
HD1(1)		2362	
↓		↓	
HD1(15)	Header controls, see subroutine INPUT	2376	
HD2(1)		2377	
↓		↓	
HD2(15)		2391	
OUTND(1)		2392	
↓	Output controls, see subroutine INPUT	↓	
OUTNO(30)		2421	
INNO(1)		2422	
↓	Input controls, see subroutine INPUT	↓	
INNO(200)		2621	
VALVE(1)		2622	
↓	Input controls, see subroutine INPUT	↓	
VALVE(200)		2821	
STPNO(1)		2822	
↓	Stop (termination) controls,	↓	
STPNO(10)	see subroutine INPUT	2831	
SUP(1)		2832	
↓	Upper bound termination locations	↓	
SUP(10)		2841	
SLO(1)	Lower bound termination	2842	
↓	locations	↓	
SLO(10)		2851	

PARAMETER	IDENTIFICATION	LOCATION IN Y ARRAY	
PLOT(1)		2852	
↓	Plot variable locations	↓	
PLOT(10)		2861	
TI	Initial time	2862	*
STOP	Termination parameters	2863	
TS		2864	
PRFR	Print control, see	2865	*
RUNN	RUN Number } Printed on each	2866	*
DATE	Date } page of printout	2867	*
DELT	Integration time step (see)	2868	
NOVAR	Number of dependent variables	2869	
NODER	Number of derivatives	2870	
NOTAB	Number of tables	2871	
LOCC(1)		2872	
↓	Locations of dependent variables	↓	
LOCC(31)	in C array	2902	
CVAR(1)	Y array locations of	2903	
↓	dependent variables	↓	
CVAR(31)		2933	
LOCD(1)	Locations of derivatives	2934	
↓	in D array	↓	
LOCD(31)		2964	
DVAR(1)	Y array locations of	2965	
↓	derivatives	↓	
DVAR(31)		2995	
KEND	Size of table array	2996	
--	Print increment when MPR=0	2997	*
PRVA	Print variable when MPR=0	2998	*
T	Time (sec)	2999	*
RE	Earth's radius, default value is 20925631 ft	3000	*
WE	Earth's spin rate, default value is $7.29211508 \times 10^{-5}$ rad/sec	3001	*
		3002	
		3003	
D	Aerodynamic reference length (ft)	3004	*
A	Aerodynamic reference area (ft^2)	3005	*
DXG		3006	*
DYG	Center of gravity location w/r	3007	*
DZG	to geometric axes, see Figure 7	3008	*
IXX		3009	*
IYY	Principal moments of inertia (slug-ft^2)	3010	*
IZZ		3011	*
MS	Vehicle mass (slug)	3012	*
H	Vehicle altitude (ft)	3013	*
TAUR	Longitude (deg)	3014	*
PSIR	Latitude (deg)	3015	*
I_{XY}		3016	*
I_{XZ}	Cross products of inertia	3017	*
I_{YZ}		3018	*

PARAMETER	IDENTIFICATION	LOCATION IN Y ARRAY
		3019
		3020
		3021
		3022
		3023
		3024
		3025
		3026
		3027
		3028
		3029
		3030
		3031
		3032
		3033
		3034
		3035
		3036
		3037
		3038
		3039
		3040
		3041
		3042
PLCT	Number of data points being plotted	3043
TITL(1)	Storage location for	3044
↓	title	↓
TITL(7)		3050
K(1)	Intermediate table	3051
↓	identification; see subroutine ITAB	↓
K(49)		3099
--		3100
--	Output array storage	↓
--		4889
KTAB(1)	Assigned array locations	4890
↓	for tabulated functions	↓
KTAB(49)		4938
		4939
		4940

APPENDIX C

IMOD1; INITIAL DIRECTION COSINE MATRIX FOR 3DOF OVER A ROTATING SPHERICAL EARTH

As mentioned previously, MOD1 contains the direction cosine matrix. Since it would be unwieldly to read in the initial values of the direction cosine matrix, IMOD1 is used to calculate these values from the initial latitude and longitude which must be read from the input cards. This transform matrix for transforming vectors from the inertial axes to the local axes is presented as:

$$[\ell_{RL}] = \begin{bmatrix} \cos\psi_R & \sin\psi_R \sin\tau_R & -\sin\psi_R \cos\tau_R \\ 0 & \cos\tau_R & \sin\tau_R \\ \sin\psi_R & -\cos\psi_R \sin\tau_R & \cos\psi_R \cos\tau_R \end{bmatrix}$$

IMOD1 Parameters

	Parameter	Units	Location in Y Array
Input			
	ψ_R	Degree	3015
	τ_R	Degree	3014
Output			
	$[\ell_{RL}]$		2000-2008

```
      SUBROUTINE IMOD1                                                  TR300100
C
C         3/10/78
C                                                                       TR300110
C     THIS ROUTINE CALCULATES THE INITIAL DIRECTION COSINE MATRICE LRL  TR300150
C     FOR 3DOF SIMULATIONS                                              TR300160
C                                                                       TR300170
C     IT REQUIRES THE FOLLOWING INITIAL CONDITIONS ON CODE 3 CONTROL    TR300180
C     CARDS                                                             TR300190
C                                                                       TR300200
C     TAUR=Y(3014) ,  LONGITUDE (DEG)
C     PSIR=Y(3015) ,  LATITUDE (DEG)
C                                                                       TR300230
      COMMON Y(4940)
      EQUIVALENCE(Y(3015),PSIR),(Y(3014),TAUR)
      EQUIVALENCE(Y(2000),LRL(1))                                       TR300260
      REAL LRL(9)                                                       TR300270
      CALL SENCOS(PSIR,SP,CP,0)                                         TR300280
      CALL SENCOS(TAUR,ST,CT,0)                                         TR300290
      LRL(1)=CP                                                         TR300300
      LRL(2)=0.                                                         TR300310
      LRL(3)=SP                                                         TR300320
      LRL(4)=SP*ST                                                      TR300330
      LRL(5)=CT                                                         TR300340
      LRL(6)=-CP*ST                                                     TR300350
      LRL(7)=-SP*CT                                                     TR300360
      LRL(8)=ST                                                         TR300370
      LRL(9)=CP*CT                                                      TR300380
      RETURN                                                            TR300390
      END                                                               TR300400
                                                                        TR300410
                                                                        TR300420
                                                                        TR300420
```

APPENDIX D

MOD1; DIRECTION COSINE MATRIX FOR 3DOF OVER A ROTATING SPHERICAL EARTH

The purpose of this module is to calculate the direction cosine matrix for a vehicle flying a 3DOF particle trajectory over a rotating spherical earth.

The longitude and latitude of the vehicle are calculated from the inertial coordinates as,

$$\tau_R = \tan^{-1} \frac{-Y_R}{Z_R}$$

$$\psi_R = \tan^{-1}\left(\frac{X_R}{\sqrt{Y_R^{\,2} + Z_R^{\,2}}}\right)$$

The direction cosine matrix which allows for transforming vectors from the inertial axes to the local axes is calculated as,

$$[\ell_{RL}] = \begin{bmatrix} \cos\psi_R & \sin\psi_R \sin\tau_R & -\sin\psi_R \cos\tau_R \\ 0 & \cos\tau_R & \sin\tau_R \\ \sin\psi_R & -\cos\psi_R \sin\tau_R & \cos\psi_R \cos\tau_R \end{bmatrix}$$

The transfer matrix $[\ell_{LR}]$ is expressed as $[\ell_{LR}] = [\ell_{RL}]^{-1}$.

MOD1 Parameters

	Parameters	Units	Location in Y Array
Input			
	X_R	feet	803
	Y_R	feet	804
	Z_R	feet	805
Output	τ_R	radian	3014
	ψ_R	radian	3015
	$[\ell_{RL}]$		2000-2008
	$[\ell_{LR}]$		2039-2047

```
      SUBROUTINE MOD1                                                   TR300930
C                                                                       TR300940
C
C     THIS ROUTINE CALCULATES THE DIRECTION COSINE MATRICE LRL FOR A    TR300980
C     3DOF SIMULATION OVER A ROTATING SPHERICAL EARTH                   TR300990
C                                                                       TR301000
C        3/10/78
C
      COMMON Y(4940)
      EQUIVALENCE(Y(803),XR),(Y(804),YR),(Y(805),ZR)                    TR301020
      EQUIVALENCE(Y(3015),PSIR),(Y(3014),TAUR)
      EQUIVALENCE(Y(2000),LRL(1))                                       TR301040
      EQUIVALENCE(Y(2039),LLR(1))                                       TR301050
      REAL LRL(9),LLR(9)                                                TR301060
      DIMENSION B(9)                                                    TR301070
C
C     LATITUDE AND LONGITUDE CALCULATED
C
10    CALL ARKTAN(-YR,ZR,TAUR,1)                                        TR301080
      CALL ARKTAN(XR,(SQRT(ZR**2+YR**2)),PSIR,1)                        TR301090
C
C     INERTIAL TO LOCAL AXES TRANSFER MATRIX
C
   20 CALL SENCOS(TAUR,STAR,CTAR,1)                                     TR301100
      CALL SENCOS(PSIR,SPSR,CPSR,1)                                     TR301110
      LRL(1)=CPSR                                                       TR301120
      LRL(2)=0.                                                         TR301130
      LRL(3)=SPSR                                                       TR301140
      LRL(4)=SPSR*STAR                                                  TR301150
      LRL(5)=CTAR                                                       TR301160
      LRL(6)=-CPSR*STAR                                                 TR301170
      LRL(7)=-SPSR*CTAR                                                 TR301180
      LRL(8)=STAR                                                       TR301190
      LRL(9)=CPSR*CTAR                                                  TR301200
      CALL MATINV(LRL,B,LLR)                                            TR301210
      RETURN                                                            TR301220
      END                                                               TR301230
                                                                        TR301240
                                                                        TR301250
                                                                        TR301260
```

APPENDIX E

IMOD2; INITIAL CONDITIONS FOR A 3DOF OR 6DOF TRAJECTORY

The purpose of this module is to calculate the initial values for the inertial coordinates and velocities of the vehicle flying over a spherical earth.

Since the initial altitude latitude, and longitude of the vehicle are specified in the input conditions, the inertial coordinates can be expressed as,

$$X_R = \sin\psi_R(h + R_E)$$

$$Y_R = -\cos\psi_R(h + R_E)\sin\tau_R$$

$$Z_R = \cos\psi_R(h + R_E)\cos\tau_R$$

Likewise, the velocity components with respect to the inertial axes can be expressed as functions of the initial inputed flight path angles and resultant velocity as,

$$\begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{Z}_R \end{bmatrix} = [\ell_{RL}]^{-1} \begin{bmatrix} V_E\cos\gamma_E\cos\gamma_H \\ -V_E\cos\gamma_E\sin\gamma_H - \omega_E(R_E + h)\cos\psi_R \\ V_E\sin\gamma_E \end{bmatrix}$$

The initial starting point of the vehicle's flight path as projected onto the earth's surface is defined as,

$$R_{\tau_{E_i}} = R_E\tau_R \left(\frac{\pi}{180}\right)$$

$$R_{\psi_{E_i}} = R_E\psi_R \left(\frac{\pi}{180}\right)$$

This IMOD serves for MOD3 and MOD9 also.

IMOD2 Parameters

Parameter	Units	Location in Y Array
Input		
h	feet	3013
R_E	feet	3000
(R_E default value = 20925631 feet)		
V_E	feet/second	610
γ_H	degree	2208
γ_E	degree	2209
ω_E	radians/sec	3001
(ω_E default value = 0.0000729211508 rad/sec)		
τ_R	degree	3014
ψ_R	degree	3015
Output		
X_R	feet	803
Y_R	feet	804
Z_R	feet	805
$\dot{X}_R$	feet/second	800
$\dot{Y}_R$	feet/second	801
$\dot{Z}_R$	feet/second	802
$R_{\tau_{E_i}}$	feet	818
$R_{\psi_{E_i}}$	feet	819

```
      SUBROUTINE IMOD2                                                  TR300430
C                                                                       TR300440
C        3/10/78
C                                                                       TR300460
C     THIS ROUTINE CALCULATES THE INITIAL VALUES OF XR,YR,ZR            TR300490
C     AND XRD, YRD, AND ZRD, AND THE INITIAL EARTH POSITIONS            TR300500
C     RTEI, AND RPEI                                                    TR300510
C
C     THE FOLLOWING INITIAL CONDITIONS ARE REQUIRED ON CODE 3
C     CONTROL CARDS
C
C     PSIR=Y(3015)   ,  LATITUDE(DEG)
C     TAUR=Y(3014)   ,  LONGITUDE(DEG)
C     H=Y(3013)   ,  ALTITUDE(FT)
C     RE=Y(3000)   ,  RADIUS OF EARTH(FT), DEFAULT VALUE = 20925631.
C     WE=Y(3001)   ,  EARTHS SPIN RATE(RAD/SEC), DEFAULT VALUE =
C       0.0000729211508
C     VE=Y(610)   ,  INITIAL VELOCITY W/R TO EARTH (FPS)
C     GAMAE=Y(2209)   ,  VELOCITY ELEVATION ANGLE (DEG)
C     GAMAH=Y(2208)   ,  VELOCITY HEADING ANGLE (DEG)
C
C
      COMMON Y(4940)
      EQUIVALENCE(Y(803),XR),(Y(804),YR),(Y(805),ZR)                    TR300560
      EQUIVALENCE(Y(2039),LLR(1)),(Y(2000),LRL(1))                      TR300570
      EQUIVALENCE(Y(3015),PSIR),(Y(3014),TAUR)
      EQUIVALENCE(Y(3001),WE)                                           TR300590
      EQUIVALENCE (Y(818),RTEI),(Y(819),RPEI)                           TR300600
      EQUIVALENCE(Y(3000),RE),(Y(3013),H)                               TR300610
      EQUIVALENCE(Y(2213),OT),(Y(2214),OP)                              TR300620
      EQUIVALENCE(Y(658),MI),(Y(3012),MS)                               TR300630
      EQUIVALENCE(Y(2208),GAMAH),(Y(2209),GAMAE)                        TR300640
      REAL LLR(9),LRL(9),LR                                             TR300650
      DIMENSION B(9)                                                    TR300660
      ENTRY IMOD3                                                       TR300665
      ENTRY IMOD9
C
C     INITIAL INERTIAL POSITION
C
      CALL SENCOS(TAUR,STAR,CTAR,0)                                     TR300670
      CALL SENCOS(PSIR,SPSR,CPSR,0)                                     TR300680
      XR=SPSR*(H+RE)                                                    TR300690
      LR=CPSR*(H+RE)                                                    TR300700
      YR=-LR*STAR                                                       TR300710
      ZR=LR*CTAR                                                        TR300720
C
C     INITIAL INERTIAL VELOCITIES
C
      CALL SENCOS(GAMAE,SE,CE,0)                                        TR300730
      CALL SENCOS(GAMAH,SH,CH,0)                                        TR300740
      Y(512)=Y(610)*CE*CH                                               TR300750
      Y(513)=-Y(610)*CE*SH                                              TR300760
      Y(514)=Y(610)*SE                                                  TR300770
      Y(513)=Y(513)-WE*(RE+H)*CPSR                                      TR300780
      CALL MATINV(LRL(1),B,LLR)                                         TR300790
      CALL MATVEC(LLR,Y(512),Y(800),0)                                  TR300800
      DO 20 I=1,3                                                       TR300860
20    Y(I+519)=Y(I+511)                                                 TR300870
C
C     INITIAL RANGES
C
      RAD=3.14159265358 9/180.                                          TR300810
```

```
      RTEI=RE*TAUR*RAD                                            TR300820
      RPEI=RE*PSIR*RAD                                            TR300830
      OT=TAUR*RAD                                                 TR300840
      OP=PSIR*RAD                                                 TR300850
      MI=MS                                                       TR300880
      RETURN                                                      TR300890
      END                                                         TR300900
                                                                  TR300910
                                                                  TR300920
```

APPENDIX F

MOD2; 3DOF PARTICLE TRAJECTORY ALONG A PROGRAMMED FLIGHT PATH

The purpose of this module is to calculate the transformation matrix for a vehicle flying along a 3DOF particle trajectory over a spherical earth where the elevation angle of the velocity vector is preprogrammed. This elevation angle, γ_E, is taken from a table of γ_E as a function of time which is to be supplied by the user as Table Array Number 10. The heading angle, γ_H, is maintained constant. The altitude is calculated as,

$$h = \sqrt{X_R^{\,2} + Y_R^{\,2} + Z_R^{\,2}} - R_E$$

The local velocities with respect to the earth are calculated as,

$$\begin{bmatrix} V_{X_{LE}} \\ V_{Y_{LE}} \\ V_{Z_{LE}} \end{bmatrix} = [\ell_{RL}] \begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{Z}_R \end{bmatrix} + \begin{bmatrix} o \\ \omega_E (R_E + h)\cos\psi_R \\ o \end{bmatrix}$$

The resultant velocity is then calculated as,

$$V_E = \sqrt{V_{X_{LE}}^{\,2} + V_{Y_{LE}}^{\,2} + V_{Z_{LE}}^{\,2}}$$

The $[\ell_{PL}]$ matrix, necessary for transforming vectors from the principal axes to the local axes is written in terms of γ_H and γ_E as,

$$[\ell_{PL}] = \begin{bmatrix} \cos\gamma_H\cos\gamma_E & \sin\gamma_H & -\cos\gamma_H\sin\gamma_E \\ -\sin\gamma_H\cos\gamma_E & \cos\gamma_H & \sin\gamma_H\sin\gamma_E \\ \sin\gamma_E & 0 & \cos\gamma_E \end{bmatrix}$$

MOD2 Parameters

	Parameter	Units	Location in Y Array
Input			
	γ_E	degree	2209
	γ_H	degree	2208
	X_R	feet	803
	Y_R	feet	804
	Z_R	feet	805
	$\dot{X}_R$	feet/second	800
	$\dot{Y}_R$	feet/second	801
	$\dot{Z}_R$	feet/second	802
	R_E	feet	3000
Output			
	$[\ell_{PL}]$		2057-2065
	h	feet	3013
	$V_{X_{1e}}$	feet/second	512
	$V_{Y_{LE}}$	feet/second	513
	$V_{Z_{LE}}$	feet/second	514
	V_E	feet/second	610

```
      SUBROUTINE MOD2
C
C           POSITION MODULE FOR A 3DOF PARTICLE TRAJ. WITH PROGRAMMED
C           PITCH ANGLE
C
C        3/10/78
C
C     GAMAE (DEG) IS TABULATED AS A FUNCTION OF TIME (SEC) IN
C        TABLE ARRAY NO. 10
C     GAMAH (DEG) IS A CONSTANT IN LOCATION Y(2208)
C                                                                       TR301280
      COMMON Y(4940)
      EQUIVALENCE  (Y(3013),H)
      EQUIVALENCE(Y(803),XR),(Y(804),YR),(Y(805),ZR)                    TR301460
      EQUIVALENCE(Y(3015),PSIR),(Y(3014),TAUR)
      EQUIVALENCE(Y(3001),WE),(Y(3000),RE)                              TR301500
      EQUIVALENCE(Y(512),VXLE),(Y(513),VYLE),(Y(514),VZLE)              TR301560
      EQUIVALENCE(Y(2999),T)                                            TR301610
      EQUIVALENCE(Y(2208),GAMAH),(Y(2209),GAMAE)                        TR301650
      EQUIVALENCE(Y(2057),LPL(1))
      EQUIVALENCE(Y(2000),LRL(1)),(Y(610),VE)
      DIMENSION U(2)
      REAL         LRL(9),LPL(9)
C
C           FLIGHT PATH ANGLES
C
   10 U(1)=T
      CALL ITAB(10,1,T,GAMAE)
      CALL SENCOS(GAMAH,SH,CH,0)                                        TR302160
      CALL SENCOS(GAMAE,SE,CE,0)                                        TR302170
C
C           PRINCIPAL TO LOCAL TRANSFER MATRIX
C
   20 LPL(1)=CH*CE                                                      TR302180
      LPL(2)=-SH*CE                                                     TR302190
      LPL(3)=SE                                                         TR302200
      LPL(4)=SH                                                         TR302210
      LPL(5)=CH                                                         TR302220
      LPL(6)=0.0                                                        TR302230
      LPL(7)=-CH*SE                                                     TR302240
      LPL(8)=SH*SE                                                      TR302250
      LPL(9)=CE                                                         TR302260
C
C           ALTITUDE
C
   30 H=SQRT(XR**2+YR**2+ZR**2)-RE                                      TR301740
C
C           LOCAL VELOCITY COMPONENTS
C
   40 CALL MATVEC(LRL(1),Y(800),Y(512),0)
      VXLE=Y(512)                                                       TR301750
      VYLE=Y(513)+WE*(RE+H)*COS(PSIR)                                   TR301760
      VZLE=Y(514)                                                       TR301770
      VE=SQRT(VXLE**2+VYLE**2+VZLE**2)
      RETURN                                                            TR302320
      END                                                               TR302330
```

APPENDIX G

MOD3; 3DOF PARTICLE TRAJECTORY WITH THRUST

The purpose of this module is to calculate the altitude, local velocity components, flight path angles, and the principal-to-local transfer matrix for a vehicle flying along a 3DOF, point-mass, ballistic trajectory.

The altitude is calculated from the inertial coordinates which have been obtained through the integration of the equations of motion;

$$h = \sqrt{X_R^{\,2} + Y_R^{\,2} + Z_R^{\,2}} - R_E$$

Since the velocity of the vehicle with respect to the inertial axes are also known from the integration of the equations of motion, the local velocities with respect to the earth's surface are calculated by transforming the inertial velocities as,

$$\begin{bmatrix} V_{X_{LE}} \\ V_{Y_{LE}} \\ V_{Z_{LE}} \end{bmatrix} = [\ell_{RL}] \begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{Z}_R \end{bmatrix} + \begin{bmatrix} o \\ \omega_E(R_E + h)\cos\psi_R \\ o \end{bmatrix}$$

The total velocity with respect to the earth is then,

$$V_E = \sqrt{V_{X_{LE}}^{\,2} + V_{Y_{LE}}^{\,2} + V_{Z_{LE}}^{\,2}}$$

The flight path angles, angles of the velocity vector with respect to the local axes, are then calculated as,

$$\gamma_H = \tan^{-1}\left(\frac{-V_{Y_{LE}}}{V_{X_{LE}}}\right),$$

$$\gamma_E = \tan^{-1}\left(\frac{V_{Z_{LE}}}{\sqrt{V_{X_{LE}}^{\,2} + V_{Y_{LE}}^{\,2}}}\right)$$

The transfer matrix for transforming a vector from the principal axes to the local axes can then be written in terms of the flight path angles as,

$$[\ell_{PL}] = \begin{bmatrix} \cos\gamma_H \cos\gamma_E & \sin\gamma_H & -\cos\gamma_H \sin\gamma_E \\ -\sin\gamma_H \cos\gamma_E & \cos\gamma_H & \sin\gamma_H \sin\gamma_E \\ \sin\gamma_E & 0 & \cos\gamma_E \end{bmatrix}$$

MOD3 Parameters

	Parameter	Units	Location in Y Array
Input			
	X_R	feet	803
	Y_R	feet	804
	Z_R	feet	805
	R_E	feet	3000
	$\dot{X}_R$	feet/second	
	$\dot{Y}_R$	feet/second	
	$\dot{Z}_R$	feet/second	
	$[\ell_{RL}]$		
	ω_E	rad/sec	
	ψ_R	radian	
Output			
	h	feet	3013
	$V_{X_{LE}}$	feet/second	512
	$V_{Y_{LE}}$	feet/second	513
	$V_{Z_{LE}}$	feet/second	514
	V_E	feet/second	610
	γ_H	degree	2208
	γ_E	degree	2209
	$[\ell_{PL}]$		2057-2065

```
      SUBROUTINE MOD3
C
C           POSITION MODULE FOR A 3DOF BALLISTIC PARTICLE TRAJ.
C                                                                       TR301300
C         3/10/78
C                                                                       TR301330
      COMMON Y(4940)
      EQUIVALENCE (Y(3013),H)
      EQUIVALENCE(Y(2057),LPL(1))
      EQUIVALENCE(Y(803),XR),(Y(804),YR),(Y(805),ZR)                    TR302530
      EQUIVALENCE(Y(3015),PSIR),(Y(3014),TAUR)
      EQUIVALENCE(Y(3001),WE),(Y(3000),RE)                              TR302570
      EQUIVALENCE(Y(512),VXLE),(Y(513),VYLE),(Y(514),VZLE)              TR302640
      EQUIVALENCE(Y(2208),GAMAH),(Y(2209),GAMAE)                        TR302730
      EQUIVALENCE(Y(2000),LRL(1)),(Y(610),VE)
      REAL LPL(9),LRL(9)
C
C           ALTITUDE
C
   10 H=SQRT(XR**2+YR**2+ZR**2)-RE                                      TR302810
C
C           LOCAL VELOCITY COMPONENTS
C
   20 CALL MATVEC(LRL(1),Y(800),Y(512),0)
      VXLE=Y(512)                                                       TR302820
      VYLE=Y(513)+WE*(RE+H)*COS(PSIR)                                   TR302830
      VZLE=Y(514)                                                       TR302840
      VE=SQRT(VXLE**2+VYLE**2+VZLE**2)
C
C           FLIGHT PATH ANGLES
C
   30 CALL ARKTAN(-VYLE,VXLE,GAMAH,0)
      ZZZ=SQRT(VXLE**2+VYLE**2)                                         TR303100
      CALL ARKTAN(VZLE,ZZZ,GAMAE,0)                                     TR303110
      CALL SENCOS(GAMAH,SH,CH,0)                                        TR303220
      CALL SENCOS(GAMAE,SE,CE,0)                                        TR303230
C
C           PRINCIPAL TO LOCAL TRANSFER MATRIX
C
   40 LPL(1)=CH*CE                                                      TR303240
      LPL(2)=-SH*CE                                                     TR303250
      LPL(3)=SE                                                         TR303260
      LPL(4)=SH                                                         TR303270
      LPL(5)=CH                                                         TR303280
      LPL(6)=0.0                                                        TR303290
      LPL(7)=-CH*SE                                                     TR303300
      LPL(8)=SH*SE                                                      TR303310
      LPL(9)=CE                                                         TR303320
C
      RETURN                                                            TR303380
      END                                                               TR303390
```

APPENDIX H

IMOD4; 6DOF INITIAL DIRECTION COSINE MATRIX

The purpose of this routine is to calculate the initial direction cosine matrices for transforming vectors from the inertial to the local to the principal axes for a 6DOF trajectory simulation over a spherical rotating earth. The inertial to local transform matrix is defined as,

$$[\ell_{RL}] = \begin{bmatrix} \cos\psi_R & \sin\psi_R \sin\tau_R & -\sin\psi_R \cos\tau_R \\ 0 & \cos\tau_R & \sin\tau_R \\ \sin\psi_R & -\cos\psi_R \sin\tau_R & \cos\psi_R \cos\tau_R \end{bmatrix}$$

and the local to principal transform matrix as

$$[\ell_{LP}] = \begin{bmatrix} \cos\varepsilon_M \cos\gamma_M & -\cos\varepsilon_M \sin\gamma_M & \sin\varepsilon_M \\ \cos\phi_M \sin\gamma_M - \sin\phi_M \sin\varepsilon_M \cos\gamma_M & \cos\phi_M \cos\gamma_M + \sin\phi_M \sin\varepsilon_M \sin\gamma_M & \sin\phi_M \cos\varepsilon_M \\ -\sin\phi_M \sin\gamma_M - \cos\phi_M \sin\varepsilon_M \cos\gamma_M & -\sin\varepsilon_M \cos\gamma_M + \cos\phi_M \sin\varepsilon_M \sin\gamma_M & \cos\phi_M \cos\varepsilon_M \end{bmatrix}$$

The angles ψ_R and τ_R are the latitude and longitude angles in degrees and the angles γ_M, ε_M, ϕ_M are the position angles of the principal axes with respect to the local angles. These angles are shown in Figures 5 and 6.

The inertial to principal axis transfer matrix can then be expressed as

$$[\ell_{RP}] = [\ell_{LP}][\ell_{RL}]$$

IMOD4 Parameters

	Parameter	Units	Location in Y Array
Input			
	τ_R	degree	3014
	ψ_R	degree	3015
	γ_M	degree	2066
	ϵ_M	degree	2067
	ϕ_M	degree	2068
Output			
	ℓ_{RL}		2000-2008
	ℓ_{RP}		2009-2017
	ℓ_{LP}		2027-2035

```
      SUBROUTINE IMOD4
C
C
C     MOD PACKAGE SIXDG. A GENERAL PURPOSE 6DOF GUIDED OR
C     UNGUIDED TRAJECTORY PROGRAM
C
C        3/10/78
C
C     THIS ROUTINE CALCULATES THE INITIAL DIRECTION COSINE MATRICES LRL,
C     LLP,LRP
C
C     IT REQUIRES THE FOLLOWING INITIAL CONDITIONS ON CODE 3 CONTROL
C     CARDS
C
C     PSIR=Y(3015)   .  LATITUDE(DEG)
C     TAUR=Y(3014)   .  LONGITUDE(DEG)
C     GAMAM=Y(2066)  .  HEADING ANGLE (DEG) OF PRINCIPAL AXES
C     EPSILM=Y(2067)  .  ELEVATION ANGLE (DEG) OF PRINCIPAL AXES
C     PHIM=Y(2068)   ,  ROLL ANGLE OF PRINCIPAL AXES
C
      COMMON Y(4940)
      EQUIVALENCE(Y(3015),PSIR),(Y(3014),TAUR)
      EQUIVALENCE(Y(2066),GAMAM),(Y(2067),EPSILM),(Y(2068),PHIM)
      EQUIVALENCE(Y(2000),LRL(1)),(Y(2009),LRP(1)),(Y(2027),LLP(1))
      REAL LRL(9),LRP(9),LLP(9)
      CALL SENCOS(PSIR,SP,CP,0)
      CALL SENCOS(TAUR,ST,CT,0)
      LRL(1)=CP
      LRL(2)=0.
      LRL(3)=SP
      LRL(4)=SP*ST
      LRL(5)=CT
      LRL(6)=-CP*ST
      LRL(7)=-SP*CT
      LRL(8)=ST
      LRL(9)=CP*CT
      CALL SENCOS(GAMAM,SG,CG,0)
      CALL SENCOS(EPSILM,SE,CE,0)
      CALL SENCOS(PHIM,SP,CP,0)
      LLP(1)=CE*CG
      LLP(2)=CP*SG-SP*SE*CG
      LLP(3)=-SP*SG-CP*SE*CG
      LLP(4)=-CE*SG
      LLP(5)=CP*CG+SP*SE*SG
      LLP(6)=-SP*CG+CP*SE*SG
      LLP(7)=SE
      LLP(8)=SP*CE
      LLP(9)=CP*CE
      CALL MATVEC(LLP(1),LRL(1),LRP(1),2)
      RETURN
      END
```

APPENDIX I

MOD4; 6DOF DIRECTION COSINE MATRIX

The purpose of this module is to calculate the direction cosine matrices ℓ_{RL}, ℓ_{RP}, and ℓ_{LP} for a 6DOF simulation of a vehicle flying over a rotating spherical earth.

First, the latitude and longitude of the vehicle are calculated as,

$$\tau_R = \tan^{-1}\left(\frac{-Y_R}{Z_R}\right)$$

$$\psi_R = \tan^{-1}\left(\frac{X_R}{\sqrt{Y_R{}^2 + Z_R{}^2}}\right)$$

The inertial to local transfer matrix can then be written as,

$$[\ell_{RL}] = \begin{bmatrix} \cos\psi_R & \sin\psi_R\sin\tau_R & -\sin\psi_R\cos\tau_R \\ 0 & \cos\tau_R & \sin\tau_R \\ \sin\psi_R & -\cos\psi_R\sin\tau_R & \cos\psi_R\cos\tau_R \end{bmatrix}$$

In order to calculate the position of the principal axes with respect to the inertial axes, it is necessary to define the direction cosine matrix as follows.

$$\begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}_P = [\ell_{RP}] \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}_R$$

where

$$\begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{bmatrix}_R$$ is a unit vector along the inertial axes

and

$$\begin{bmatrix} \vec{j} \\ \vec{i} \\ \vec{k} \end{bmatrix}_P \quad \text{is a unit vector along the principal axes}$$

and

$$[\ell_{RP}] = \begin{bmatrix} \ell_{11} & \ell_{12} & \ell_{13} \\ \ell_{21} & \ell_{22} & \ell_{23} \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix}$$

When the above are expanded, the equations look as follows:

$$\vec{i}_P = \ell_{11}\,\vec{i}_R + \ell_{12}\,\vec{j}_R + \ell_{13}\,\vec{k}_R$$

$$\vec{j}_P = \ell_{21}\,\vec{i}_R + \ell_{22}\,\vec{j}_R + \ell_{23}\,\vec{k}_R$$

$$\vec{k}_P = \ell_{31}\,\vec{i}_R + \ell_{32}\,\vec{j}_R + \ell_{33}\,\vec{k}_R$$

If these are now differentiated with respect to time the following equations result:

$$\dot{\vec{i}}_P = \dot{\ell}_{11}\,\vec{i}_R + \dot{\ell}_{12}\,\vec{j}_R + \dot{\ell}_{13}\,\vec{k}_R$$

$$\dot{\vec{j}}_P = \dot{\ell}_{21}\,\vec{i}_P + \dot{\ell}_{22}\,\vec{j}_R + \dot{\ell}_{23}\,\vec{k}_R$$

$$\dot{\vec{k}}_P = \dot{\ell}_{31}\,\vec{i}_R + \dot{\ell}_{32}\,\vec{j}_R + \dot{\ell}_{33}\,\vec{k}_R$$

If the following relationships are substituted into the above equations

$$\dot{\vec{i}}_P = \vec{\omega} \times \vec{i}_P = r\vec{j}_P - q\,\vec{k}_P$$

$$\dot{\vec{j}}_P = \vec{\omega} \times \vec{j}_P = p\vec{k}_P - r\,\vec{i}_P$$

$$\dot{\vec{k}}_P = \vec{\omega} \times \vec{k}_P = q\,\vec{i}_P - P\,\vec{j}_P$$

and the individual components are separated the following nine differential equations are formed.

$$\dot{\ell}_{11} = r\ell_{21} - q\ell_{31}$$
$$\dot{\ell}_{21} = p\ell_{31} - r\ell_{11}$$
$$\dot{\ell}_{31} = q\ell_{11} - p\ell_{21}$$
$$\dot{\ell}_{12} = r\ell_{22} - q\ell_{32}$$
$$\dot{\ell}_{22} = p\ell_{32} - r\ell_{12}$$
$$\dot{\ell}_{32} = q\ell_{12} - p\ell_{22}$$
$$\dot{\ell}_{18} = r\ell_{23} - q\ell_{33}$$
$$\dot{\ell}_{23} = p\ell_{33} - r\ell_{13}$$
$$\dot{\ell}_{33} = q\ell_{13} - p\ell_{23}$$

These nine equations can then be integrated numerically in order to define the individual elements of the direction cosine matrix ℓ_{RP}. The other matrices can then be calculated as,

$$[\ell_{LR}] = [\ell_{RL}]^{-1}$$

$$[\ell_{LP}] = [\ell_{RP}][\ell_{LR}]$$

$$[\ell_{PL}] = [\ell_{LP}]^{-1}$$

MOD4 Parameters

	Parameter	Units	Location in Y Array
Input			
	X_R	ft	803
	Y_R	ft	804
	Z_R	ft	805
	P	rad/sec	806
	q	rad/sec	807
	r	rad/sec	808
	$[\ell_{RP}]$		2009-2017
Output			
	τ_R	deg	3014
	ψ_R	deg	3015
	$[\ell_{RL}]$		2000-2008
	$[\ell_{LP}]$		2027-2035
	$[\ell_{LR}]$		2039-2047
	$[\ell_{PL}]$		2057-2065

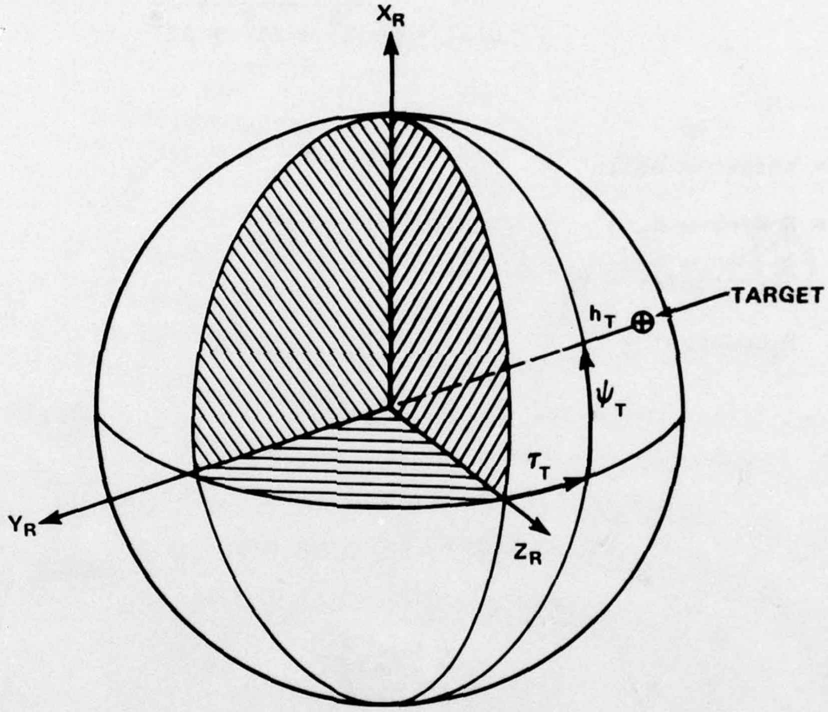
```
      SUBROUTINE MOD4                                                   SIX01260
C                                                                       SIX01270
C         3/10/78
C                                                                       SIX01290
C     MOD PACKAGE SIXDG. A GENERAL PURPOSE 6DOF GUIDED OR               SIX01300
C     UNGUIDED TRAJECTORY PROGRAM                                       SIX01310
C                                                                       SIX01320
C     THIS ROUTINE CALCULATES THE DIRECTION COSINE MATRICES LRL,LRP,    SIX01330
C     LLP FOR A 6DOF SIMULATION OVER A ROTATING SPHERICAL EARTH         SIX01340
C                                                                       SIX01350
C     NOTE"  THE LRPD(9) MATRIX ELEMENTS ARE DERIVATIVES AND MUST BE    SIX01360
C     IDENTIFIED ON CODE 6 AND 7 CONTROL CARDS                          SIX01370
C                                                                       SIX01380
      COMMON Y(4940)                                                    SIX01390
      EQUIVALENCE(Y(803),XR),(Y(804),YR),(Y(805),ZR)                    SIX01400
      EQUIVALENCE(Y(2036),PHIL),(Y(2037),THETAL),(Y(2038),PSIL)         SIX01410
      EQUIVALENCE(Y(3014),TAUR),(Y(3015),PSIR)                          SIX01420
      EQUIVALENCE(Y(2000),LRL(1)),(Y(2009),LRP(1)),(Y(2018),LRPD(1))    SIX01430
      EQUIVALENCE(Y(806),PP),(Y(807),QP),(Y(808),RP)                    SIX01440
      EQUIVALENCE(Y(2027),LLP(1)),(Y(2039),LLR(1)),(Y(2057),LPL(1))     SIX01450
      REAL LRL(9),LRPD(9),LRP(9),LLP(9),LLR(9),LPL(9)                   SIX01480
      DIMENSION B(9)                                                    SIX01490
      DEG=180./3.141592653589                                           SIX01500
C
C     LATITUDE AND LONGITUDE
C
   10 CALL ARKTAN(-YR,ZR,TAUR,0)                                        SIX01510
      CALL ARKTAN(XR,(SQRT(ZR**2+YR**2)),PSIR,0)                        SIX01520
   20 CALL SENCOS(TAUR,STAR,CTAR,0)                                     SIX01530
      CALL SENCOS(PSIR,SPSR,CPSR,0)                                     SIX01540
C
C     INERTIAL TO LOCAL, LOCAL TO PRINCIPAL, AND INERTIAL TO
C     PRINCIPAL TRANSFER MATRICES
C
      LRL(1)=CPSR                                                       SIX01550
      LRL(2)=0.                                                         SIX01560
      LRL(3)=SPSR                                                       SIX01570
      LRL(4)=SPSR*STAR                                                  SIX01580
      LRL(5)=CTAR                                                       SIX01590
      LRL(6)=-CPSR*STAR                                                 SIX01600
      LRL(7)=-SPSR*CTAR                                                 SIX01610
      LRL(8)=STAR                                                       SIX01620
      LRL(9)=CPSR*CTAR                                                  SIX01630
30    LRPD(1)=RP*LRP(2)-QP*LRP(3)                                       SIX01640
      LRPD(2)=PP*LRP(3)-RP*LRP(1)                                       SIX01650
      LRPD(3)=QP*LRP(1)-PP*LRP(2)                                       SIX01660
      LRPD(4)=RP*LRP(5)-QP*LRP(6)                                       SIX01670
      LRPD(5)=PP*LRP(6)-RP*LRP(4)                                       SIX01680
      LRPD(6)=QP*LRP(4)-PP*LRP(5)                                       SIX01690
      LRPD(7)=RP*LRP(8)-QP*LRP(9)                                       SIX01700
      LRPD(8)=PP*LRP(9)-RP*LRP(7)                                       SIX01710
      LRPD(9)=QP*LRP(7)-PP*LRP(8)                                       SIX01720
      CALL MATINV(LRL,B,LLR)                                            SIX01730
40    CALL MATVEC(LRP,LLR,LLP,2)                                        SIX01740
C
C     ORIENTATION OF PRINCIPAL AXES W/R TO LOCAL AXES
C
      CALL ARKTAN(-LLP(7),LLP(1),THETAL,0)                              SIX01750
      IF(THETAL)  43,44,44                                              SIX01760
43    THETAL=THETAL+360.                                                SIX01770
44    CONTINUE                                                          SIX01780
      CALL ARKTAN(-LLP(6),LLP(5),PHIL,0)                                SIX01790
```

```
      IF(PHIL) 46,48,48                                              SIX01800
46    PHIL=PHIL+360.                                                 SIX01810
48    CONTINUE                                                       SIX01820
      CALL SENCOS(PHIL,SPH,CPH,0)                                    SIX01830
      PSIL=ASIN(LLP(4))*DEG                                          SIX01840
50    CALL MATINV(LLP,B,LPL)                                         SIX01850
      RETURN                                                         SIX01860
      END                                                            SIX01870
```

APPENDIX J

MOD5; TARGET MODULE

This module provides for the locating of a target with respect to the vehicle. The target can be either fixed with respect to the earth or moving at a constant velocity. The initial position of the target is given by the initial altitude, h_T, latitude, ψ_{Ti}, and longitude, τ_{Ti}. The velocities are presented as vertical, $\dot{R}_T$, longitudinal, V_{T_τ}; and latitudinal, V_{T_ψ}. These relationships are shown in the following sketch:



The inertial coordinates of the target are calculated as,

$$X_{TR} = R_T \sin\psi_T$$

$$Y_{TR} = -R_T \cos\psi_T \sin\tau_T$$

$$Z_{TR} = R_T \cos\psi_T \cos\tau_T$$

where

$$R_T = R_e + h_{T_i} + \dot{R}_T(t - t_i)$$

$$\psi_T = \psi_{T_i} + \frac{V_{T_{\psi i}}}{R_T}\left(\frac{180}{\pi}\right)(t - t_i)$$

$$\tau_T = \tau_{T_i} + \frac{V_{T_{\tau i}}}{R_T \cos\psi}\left(\frac{180}{\pi}\right)(t - t_i) + \omega_E\left(\frac{180}{\pi}\right)(t - t_i)$$

The distance between the target and the vehicle are expressed as,

$$DIST = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}$$

where,

Δ = target-missile

$\Delta X = R_T \sin\psi - X_R$

$\Delta Y = -R_T \cos\psi \sin\tau - Y_R$

$\Delta Z = R_T \cos\psi \cos\tau - Z_R$

MOD5 Parameters

Parameter		Units	Location in Y Array
Input			
	h_{T_i}	ft	1
	τ_{T_i}	deg	2
	ψ_{T_i}	deg	3
	V_{T_τ}	fps	4
	V_{T_ψ}	fps	5
	$\dot{R}_T$	fps	6
	R_E	ft	3000
	ω_E	rad/sec	3001
	t_i	sec	2862
	X_R	ft	803
	Y_R	ft	804
	Z_R	ft	805
Output			
	X_{TR}	ft	10
	Y_{TR}	ft	11
	Z_{TR}	ft	12
	ΔX	ft	13
	ΔY	ft	14
	ΔZ	ft	15
	DIST	ft	19

```
      SUBROUTINE MOD5                                              SIX01910
C                                                                  SIX01920
C        3/10/78
C                                                                  SIX01940
C     TARGET FIXED OR MOVING AT A CONSTANT VELOCITY                SIX01950
C                                                                  SIX01960
C     THE REQUIRED INPUTS ARE.                                     SIX01970
C     HTI = INITIAL HEIGHT OF THE TARGET (FT)                      SIX01980
C     TAUTI = INITIAL LONGITUDE (DEG)                              SIX01990
C     PSITI = INITIAL LATITUDE (DEG)                               SIX02000
C     VTTAU = LONGITUDINAL VELOCITY OF TARGET (FPS)                SIX02010
C     VTPSI = LATITUDINAL VELOCITY OF TARGET (FPS)                 SIX02020
C     RTDOT = VERTICAL VELOCITY OF TARGET (FPS)                    SIX02030
C                                                                  SIX02040
      COMMON Y(4940)                                               SIX02050
      EQUIVALENCE(Y(1),HTI),(Y(2),TAUTI),(Y(3),PSITI)              SIX02060
      EQUIVALENCE(Y(4),VTTAU),(Y(5),VTPSI),(Y(6),RTDOT)            SIX02070
      EQUIVALENCE(Y(3000),RE),(Y(3001),WE)                         SIX02080
      EQUIVALENCE(Y(2868),DELT),(Y(2999),T)                        SIX02090
      EQUIVALENCE(Y(2862),TI)                                      SIX02100
      EQUIVALENCE(Y(7),RT),(Y(8),TAU),(Y(9),PSI)                   SIX02110
      EQUIVALENCE(Y(10),XTR),(Y(11),YTR),(Y(12),ZTR)               SIX02120
      EQUIVALENCE(Y(13),DX),(Y(14),DY),(Y(15),DZ)                  SIX02130
      EQUIVALENCE(Y(803),XR),(Y(804),YR),(Y(805),ZR)               SIX02140
      EQUIVALENCE(Y(19),DIST)                                      SIX02150
C
C     INERTIAL COORDINATES OF TARGET
C
      DEG=180./3.14159265358979                                    SIX02160
      RT=RE+HTI+RTDOT*(T-TI)                                       SIX02170
      PSI=PSITI+(VTPSI/RT)*DEG*(T-TI)                              SIX02180
      CALL SENCOS(PSI,SP,CP,0)                                     SIX02190
      TAU=TAUTI+(VTTAU/(RT*CP))*DEG*(T-TI)                         SIX02200
      TAU=TAU+WE*(T-TI)*DEG                                        SIX02210
      CALL SENCOS(TAU,ST,CT,0)                                     SIX02220
      XTR=RT*SP                                                    SIX02230
      YTR=-RT*CP*ST                                                SIX02240
      ZTR=RT*CP*CT                                                 SIX02250
C
C     INERTIAL DISTANCES BETWEEN TARGET AND MISSILE
C
      DX=XTR-XR                                                    SIX02260
      DY=YTR-YR                                                    SIX02270
      DZ=ZTR-ZR                                                    SIX02280
      DIST=SQRT(DX**2+DY**2+DZ**2)                                 SIX02290
      RETURN                                                       SIX02300
      END                                                          SIX02310
```

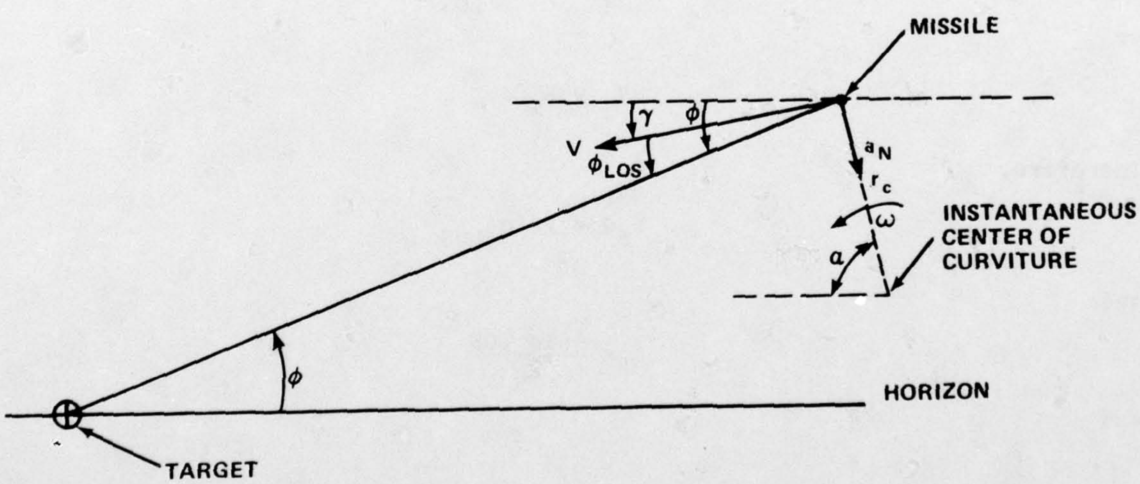
APPENDIX K

MOD6; PROPORTIONAL NAVIGATION SEEKER MODULE

This module takes the relative locations of the target and missile, converts these linear displacements into angular relationships and utilizing the laws of proportional navigation calculates error signals suitable for a control system. The definition of proportional navigation is that the angular rate of the vehicle should be proportional to the rate of change of the line-of-sight angle. In order to rotate the vehicle, or more appropriately its velocity vector, it is necessary to generate an acceleration at right angles to the velocity vector. Based on the turning radius of the vehicle,

$$a_N = \frac{V^2}{r_c}$$

where the following sketch shows the definitions of the parameters.



Since,

$$\vec{V} = \vec{\omega} \times r_c$$

$$\dot{\alpha} = - \dot{\gamma}$$

$$\omega \equiv -\dot{\alpha} = \dot{\gamma}$$

the radius of curvature is

$$r_c = V/\dot{\gamma}$$

The normal acceleration then becomes

$$a_N = V \dot{\gamma}$$

or since the definition of proportional navigation is

$$\dot{\gamma} \equiv K \dot{\phi}$$

$$a_N = VK \dot{\phi}$$

The velocity component of the missile at right angles to the line-of-sight can be defined in two ways.

$$V_{\perp} = V\sin\phi_{LOS}$$

or

$$V_{\perp} = R_S \dot{\phi}$$

Therefore,

$$R_S\dot{\phi} = V\sin\phi_{LOS}$$

and

$$\dot{\phi} = \frac{V_{\sin\ LOS}}{R_S}$$

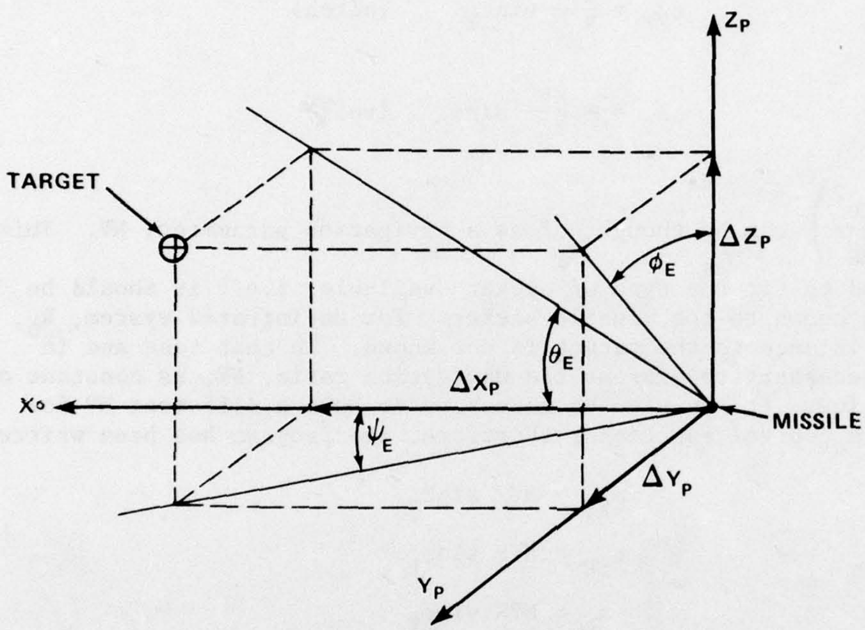
and

$$a_N = \frac{KV^2}{R_S} \sin\phi_{LOS}$$

The relative locations between the target and the missile with respect to the inertial axes were calculated in the Target Module. These "position errors" can be transformed into "body" related displacements by,

$$\begin{bmatrix} \Delta X_P \\ \Delta Y_P \\ \Delta Z_P \end{bmatrix} = [\ell_{RP}] \begin{bmatrix} \Delta X_R \\ \Delta Y_R \\ \Delta Z_R \end{bmatrix}$$

where



The angular relationships can then be calculated as

$$\psi_E = \tan^{-1}\left(\frac{\Delta Y_P}{\Delta X_P}\right)$$

$$\theta_E = \tan^{-1}\left(\frac{\Delta Z_P}{\Delta X_P}\right)$$

$$\phi_E = \tan^{-1}\left(\frac{\Delta Y_P}{\Delta Z_P}\right)$$

It is now possible to express the required normal accelerations in terms of the angular displacements. For a bi-planar control system these would be,

$$a_{ZP} = \frac{KV^2}{R_S} \sin\theta_E$$

$$a_{YP} = \frac{KV^2}{R_S} \sin\psi_E$$

If the vehicle had only planar lift and roll capability these would be,

$$a_{ZP} = \frac{KV^2}{R_S} \sin\theta_E \qquad \text{(pitch)}$$

$$a_L = -\frac{KV^2}{R_S} \sin\phi_E \qquad \text{(roll)}$$

The factor $\left(\frac{KV^2}{R_S}\right)$ can be thought of as a navigation parameter, NV. This is usually tailored to fit the type of seeker available; i.e., it should be expressed in terms known to the missile seeker. For an infrared system, R_S, or the remaining distance to the target is not known. In that case and in others it may be necessary to express the navigation ratio, NV, as constant or as a function of time. It may also be necessary to have a different NV for each of the missile control functions; therefore, the program has been written as,

$$a_{ZP} = NVP \sin\theta_E$$

$$a_{YP} = NVP \sin\psi_E$$

$$a_L = NVR \sin\phi_E$$

Each user will have to program the navigation ratios to suit his particular system.

MOD6 Parameters

	Parameter	Units	Location in Y Array
Input			
	ΔX	ft	13
	ΔY	ft	14
	ΔZ	ft	15
	NVP	ft/sec^2	126
	NVY	ft/sec^2	127
	NVR	ft/sec^2	128
Output			
	AZP	ft/sec^2	102
	AYP	ft/sec^2	101
	AL	ft/sec^2	100
	ϕ_E	rad	123
	θ_E	rad	124
	ψ_E	rad	125

```
      SUBROUTINE MOD6
C
C        3/10/78
C
C     PROPORTIONAL NAVIGATION
C
C     THIS PROGRAM TAKES THE TARGET DISPLACEMENTS, CALCULATES THE
C     ANGULAR DISPLACEMENT, AND THEN PROCESSES IT IN ORDER TO ARRIVE
C     AT ERROR SIGNALS FOR THE AUTOPILOT
C
      COMMON Y(4940)
      EQUIVALENCE(Y(2009),LRP)
      EQUIVALENCE(Y(126),NVP),(Y(127),NVY),(Y(128),NVR)
      EQUIVALENCE(Y(100),AL),(Y(101),AYP),(Y(102),AZP)
      EQUIVALENCE(Y(123),PHIE),(Y(124),THETAE),(Y(125),PSIE)
      REAL LRP(9),NVP,NVY,NVR
      CALL MATVEC(LRP,Y( 13),Y(100),0)
      CALL ARKTAN(Y(101),Y(102),PHIE,1)
      CALL ARKTAN(Y(102),Y(100),THETAE,1)
      CALL ARKTAN(Y(101),Y(100),PSIE,1)
C.....NOTE...............................
C     PROGRAM THE NAVIGATION RATIOS TO SUIT YOUR MISSILE
      NVP=Y(126)
      NVY=Y(127)
      NVR=Y(128)
C..............................................
      AZP=NVP*SIN(THETAE)
      AYP=NVY*SIN(PSIE)
      AL=NVR*SIN(PHIE)
      RETURN
      END
```

APPENDIX L

MOD7; AUTOPILOT/CONTROL MODULE

This module is representative of an autopilot or control system for a 6DOF simulation. In this elementary example, it is assumed that the vehicle has a bi-planar control system and that the vehicle does not roll. This module demonstrates how a lead-lag network can be incorporated for handling the seeker error signals, how the missile heave and pitching motion can be incorporated, and how the actuator dynamics can be included. This module has just the two channels, one for pitch and the other for yaw. The easiest way to describe the system is to refer to Figures L-1 and L-2. The basic input to this module would consist of two error signals received from the seeker. A positive error signal in the "Z" channel arriving from the seeker is calling for a correction in the missile attitude such that would cause the vehicle to be displaced in the direction of the positive Z_p axis of the vehicle. A similar arrangement exists for the yaw channel.

AD-A073 901 NAVAL SURFACE WEAPONS CENTER WHITE OAK LAB SILVER SP--ETC F/G 16/2
MODIFLY: A MODULAR MULTI-DEGREE-OF-FREEDOM TRAJECTORY PROGRAM.(U)
MAR 79 J E HOLMES
UNCLASSIFIED NSWC/WOL/TR 78-59

2 OF 2
AD
A073901

END
DATE
FILMED
10-79
DDC

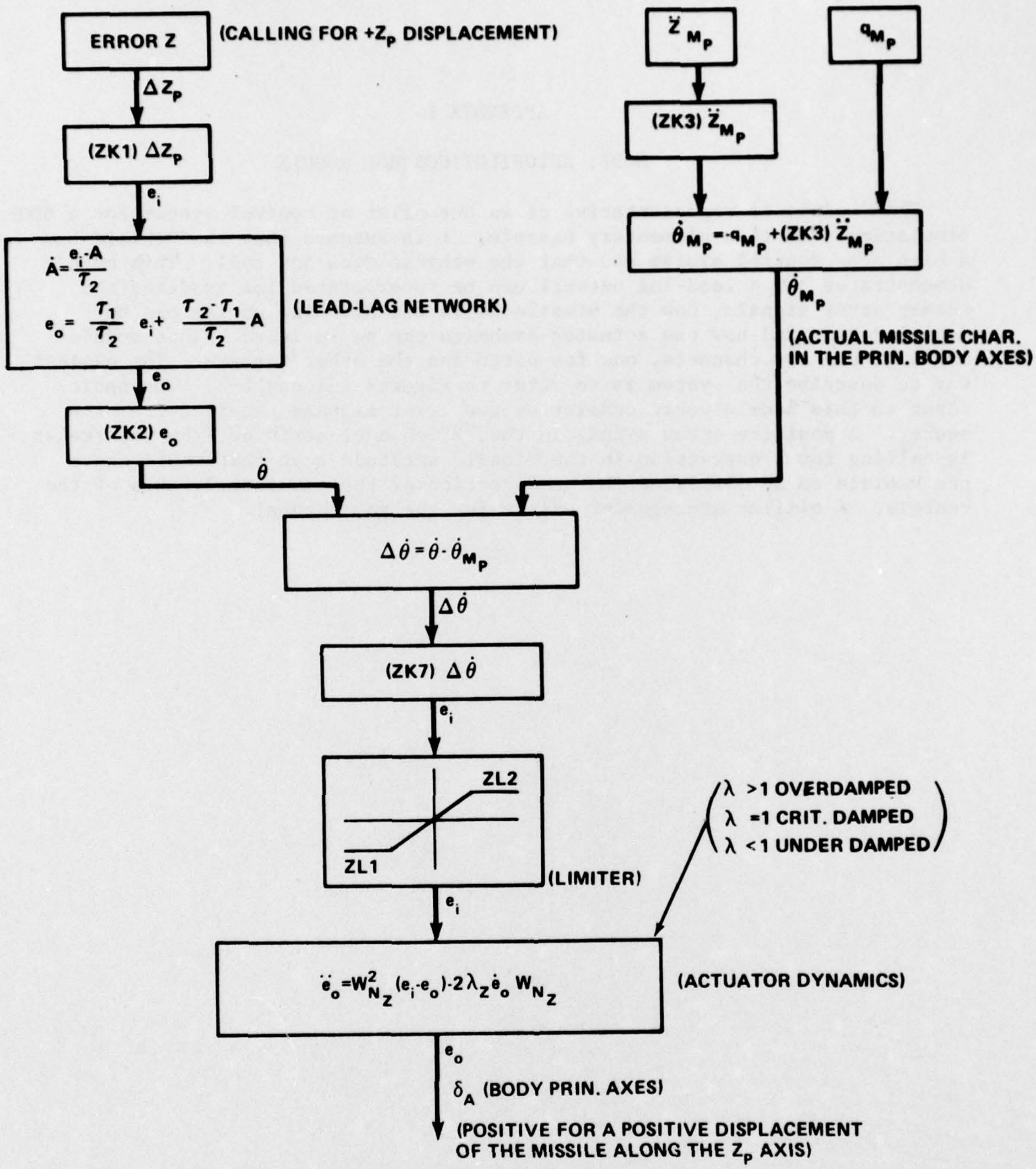


FIGURE L-1 Z CONTROL CHANNEL

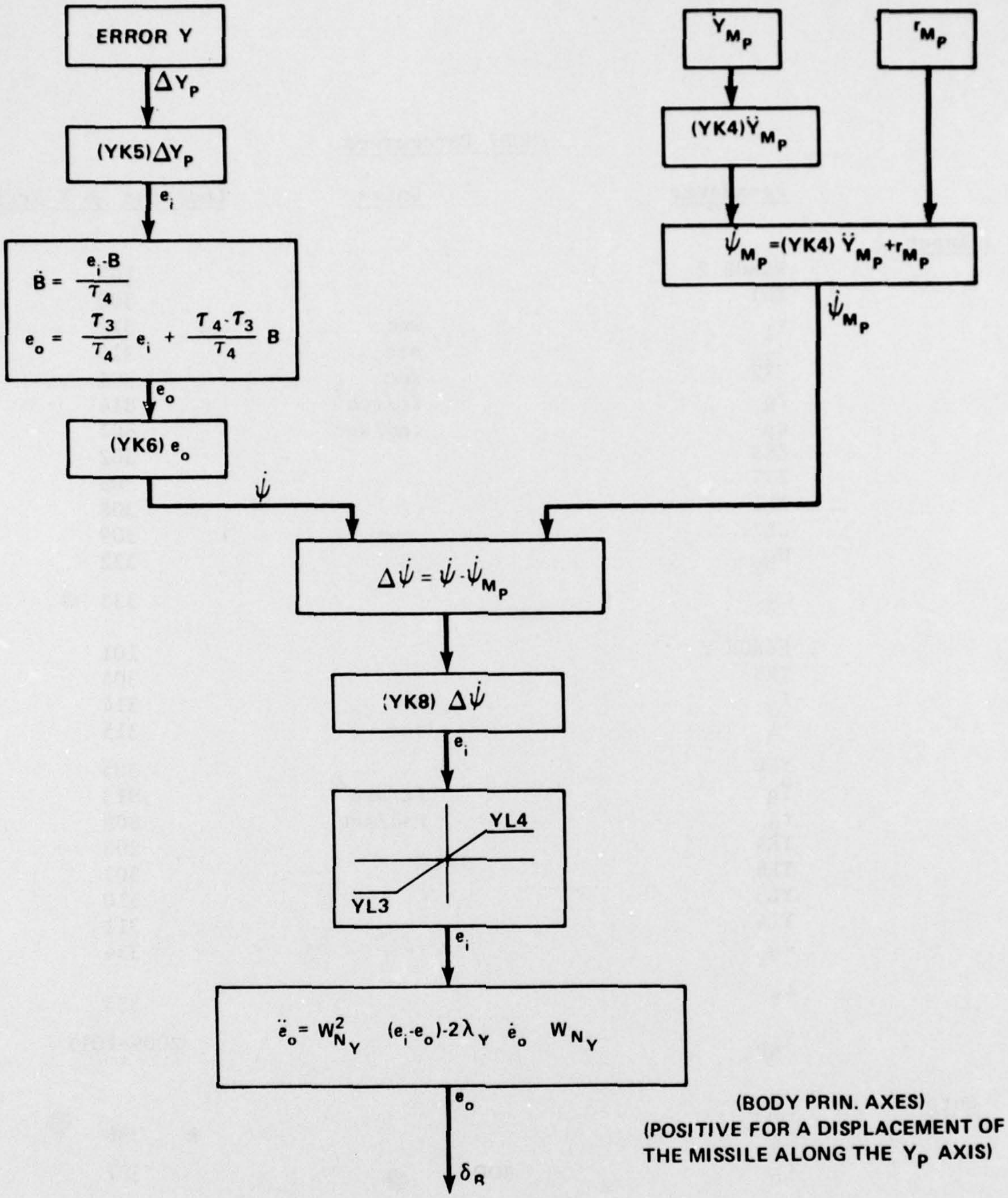


FIGURE L-2 Y CONTROL CHANNEL

MOD7 Parameters

	Parameter	Units	Location in Y Array
Input			
	ERROR Z		102
	ZK1		300
	τ_1	sec	312
	τ_2	sec	313
	ZK2	sec	301
	$\ddot{Z}_R$	ft/sec^2	814
	q_P	rad/sec	807
	ZK3		302
	ZK7		306
	ZL1		308
	ZL2		309
	W_{N_Z}		332
	λ_Z		333
	ERROR Y		101
	YK5		304
	τ_3	sec	314
	τ_4	sec	315
	YK6		305
	$\ddot{Y}_R$	ft/sec^2	813
	r_P	rad/sec	808
	YK4		303
	YK8		307
	YL3		310
	YL4		311
	ω_{N_Y}		334
	λ_Y		335
	ℓ_{RP}		2009-2017
Output			
	δ_A		386
	δ_B		387

```
      SUBROUTINE MOD7                                                   S1100450
C                                                                       S1100480
C                                                                       S1100490
C     AUTOPILOT BASE1,BI-PLANAR CONTROL DEFLECTIONS                     S1100500
C                                                                       S1100460
C       3/10/78
C                                                                       S1100510
C     NOTE THAT THIS PROGRAM REQUIRES CODE 6 AND CODE 7 CONTROL
C     CARDS FOR A,AD,B,BD,EOZ,EOZD,EOZDD,EOY,EOYD,EOYDD
C
      COMMON Y(4940)
      EQUIVALENCE(Y(100),ERRORX),(Y(101),ERRORY),(Y(102),ERRORZ)        S1100530
      EQUIVALENCE(Y(300),ZK1),(Y(301),ZK2),(Y(302),ZK3)                 S1100540
      EQUIVALENCE(Y(303),YK4),(Y(304),YK5),(Y(305),YK6)                 S1100550
      EQUIVALENCE(Y(306),ZK7),(Y(307),YK8),(Y(308),ZL1)                 S1100560
      EQUIVALENCE(Y(309),ZL2),(Y(310),YL3),(Y(311),YL4)                 S1100570
      EQUIVALENCE(Y(312),T1),(Y(313),T2),(Y(314),T3),(Y(315),T4)        S1100580
      EQUIVALENCE(Y(332),WNZ),(Y(333),LZ),(Y(334),WNY),(Y(335),LY)      S1100590
      EQUIVALENCE(Y(316),AD),(Y(317),A)                                 S1100600
      EQUIVALENCE(Y(318),BD),(Y(319),B)                                 S1100610
      EQUIVALENCE(Y(806),PP),(Y(807),QP),(Y(808),RP)                    S1100620
      EQUIVALENCE(Y(812),XDD),(Y(813),YDD),(Y(814),ZDD)                 S1100630
      EQUIVALENCE(Y(2000),LRL(1)),(Y(2009),LRP(1))                      S1100640
      EQUIVALENCE(Y(359),THED),(Y(360),PSID)                            S1100650
      EQUIVALENCE(Y(329),EOZDD),(Y(330),EOZD),(Y(331),EOZ)              S1100660
      EQUIVALENCE(Y(336),EOYDD),(Y(337),EOYD),(Y(338),EOY)              S1100670
      EQUIVALENCE(Y(380),PMP),(Y(381),QMP),(Y(382),RMP)                 S1100680
      EQUIVALENCE(Y(386),DA),(Y(387),DB)                                S1100690
      EQUIVALENCE(Y(2057),LPL(1))                                       S1100700
      REAL LRL(9),LRP(9),LPL(9),LZ,LY                                   S1100710
C
C     PITCH CHANNEL
C
10    EI=ZK1*ERRORZ                                                     S1100720
      AD=(EI-A)/T2                                                      S1100730
      EO=(T1/T2)*EI+((T2-T1)/T2)*A                                      S1100740
   20 THED=ZK2*EO                                                       S1100750
C
C     YAW CHANNEL
C
30    EI=YK5*ERRORY                                                     S1100760
      BD=(EI-B)/T4                                                      S1100770
      EO=(T3/T4)*EI+((T4-T3)/T4)*B                                      S1100780
   40 PSID=YK6*EO                                                       S1100790
C
C     MISSILE ANGULAR RATES IN PRINCIPAL AXES
C
      CALL MATVEC(LRP,Y(812),Y(383),0)                                  S1100800
   50 PMP=PP                                                            S1100810
      QMP=-QP+ZK3*Y(385)                                                S1100820
      RMP=RP+YK4*Y(384)                                                 S1100830
  100 Y(323)=0.                                                         S1100840
      Y(324)=(THED-QMP)*ZK7                                             S1100850
      Y(325)=(PSID-RMP)*YK8                                             S1100860
C
C     LIMITS ON ACTUATOR SIGNALS
C
      IF(Y(324).GE.ZL2) Y(324)=ZL2                                      S1100870
      IF(Y(324).LE.ZL1) Y(324)=ZL1                                      S1100880
      IF(Y(325).GE.YL4) Y(325)=YL4                                      S1100890
      IF(Y(325).LE.YL3) Y(325)=YL3                                      S1100900
C
```

```
C       ACTUATOR DYNAMICS
C
        EOYDD=(WNY**2)*(Y(325)-EOY)-2.0*LY*EOYD*WNY                     S1100910
        EOZDD=(WNZ**2)*(Y(324)-EOZ)-2.0*LZ*EOZD*WNZ                     S1100920
        DA=EOZ                                                          S1100930
        DB=EOY                                                          S1100940
        RETURN                                                          S1100950
        END                                                             S1100960
```

APPENDIX M

MOD8; 6DOF FORCE AND MOMENT MODULE

The purpose of this module is to calculate the external forces and moments acting on a vehicle in a 6DOF simulation. It provides for nonlinear aerodynamics, winds, thrust, and relatively small control moments.

The velocity of the vehicle with respect to the earth is calculated by transforming the inertial velocities as follows,

$$\begin{bmatrix} V_{X_{LE}} \\ V_{Y_{LE}} \\ V_{Z_{LE}} \end{bmatrix} = [\ell_{RL}] \begin{bmatrix} \dot{X}_r \\ \dot{Y}_R \\ \dot{Z}_R \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_E (R_E + h) \cos\psi_R \\ 0 \end{bmatrix}$$

Winds are then introduced in tabular form as a function of altitude. The wind velocity, V_W is tabulated as a function of altitude in Table Array No. 3, and the heading angle of the wind is tabulated in Table Array No. 4 (see Figure 8). The velocity of the vehicle with respect to the air is then calculated as,

$$\begin{bmatrix} V_{A_{XL}} \\ V_{A_{YL}} \\ V_{A_{ZL}} \end{bmatrix} = \begin{bmatrix} V_{Y_{LE}} \\ V_{Y_{LE}} \\ V_{Z_{LE}} \end{bmatrix} - \begin{bmatrix} V_W \cos(A_W) \\ V_W \sin(A_W) \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} V_{A_{XP}} \\ V_{A_{YP}} \\ V_{A_{ZP}} \end{bmatrix} = [\ell_{LP}] \begin{bmatrix} V_{A_{XL}} \\ V_{A_{YL}} \\ V_{A_{ZL}} \end{bmatrix}$$

If the geometric axes in which the aerodynamic data were measured are skewed with respect to the principal axes, the velocity can be further transformed as

$$\begin{bmatrix} V_{A_{XG}} \\ V_{A_{YG}} \\ V_{A_{ZG}} \end{bmatrix} = [\ell_{PG}] \begin{bmatrix} V_{A_{XP}} \\ V_{A_{YP}} \\ V_{A_{ZP}} \end{bmatrix}$$

and the total velocity of the vehicle with respect to the air is,

$$V_A = \sqrt{V_{A_{XG}}{}^2 + V_{A_{YG}}{}^2 + V_{A_{ZG}}{}^2}$$

In order to calculate the Mach number and dynamic pressure it is necessary to calculate the flow peroperties such as speed of sound and density. These values are derived from the 1969 U.S. Standard Atmospheric Tables.

The angles between the body and the flow vector are defined as:

$$\left.\begin{aligned} \alpha &= \tan^{-1}\left[\frac{V_{A_{ZG}}}{V_{A_{XG}}}\right] \\ \beta &= \tan^{-1}\left[\frac{V_{A_{YG}}}{V_{Z_{XG}}}\right] \\ \phi_A &= \tan^{-1}\left[\frac{V_{A_{YG}}}{V_{A_{ZG}}}\right] \\ \bar{\alpha} &= \tan^{-1}\left[\frac{\sqrt{V_{A_{YG}}{}^2 + V_{A_{ZG}}{}^2}}{V_{A_{XG}}}\right] \end{aligned}\right\} \text{ see Figure 9}$$

The aerodynamic forces and moments are entered into the program through a set of tables. These tables are as follows:

AERO COEFF.	TABLE ARRAY NO.
$C_x(M, \bar{\alpha})$	12
$C_y(M, \bar{\alpha}, \phi_A)$	13
$C_z(M, \bar{\alpha}, \phi_A)$	14
$C_{y_p}(M, \bar{\alpha})$	15
$C_\ell(M, \bar{\alpha}, \phi_A)$	16
$C_{\ell_p}(M, \bar{\alpha})$	17
$C_{\ell_\delta}(M, \bar{\alpha})$	18
$C_m(M, \bar{\alpha}, \phi_A)$	19
$C_n(M, \bar{\alpha}, \phi_A)$	20
$C_{m_q}(M, \bar{\alpha})$	21
$C_{n_p}(M, \bar{\alpha})$	22

The forces along the vehicle geometric axes can then be defined as:

$$\begin{bmatrix} F_{XG} \\ F_{YG} \\ F_{ZG} \end{bmatrix} = \left(\frac{1}{2}\rho V_A^2\right) *(A)* \begin{bmatrix} C_{XG} \\ C_{YG} \\ C_{ZG} \end{bmatrix} - \begin{bmatrix} \text{THRUST} \\ 0 \\ 0 \end{bmatrix}$$

where

$$C_{XG} = C_x$$

$$C_{YG} = C_y \cos\phi_A + C_y \sin\phi_A + C_{y_p} \frac{pd}{2V_A} \cos\phi_A$$

$$C_{ZG} = C_z \cos\phi_A - C_y \sin\phi_A - C_{y_p} \frac{pd}{2V_A} \sin\phi_A$$

The aerodynamic moment coefficients are written as

$$C_{LG} = C_{\ell} + C_{\ell_p} \frac{pd}{2V_A} + C_{\ell_\delta} \delta$$

$$C_{MG} = C_m \cos\phi_A + C_n \sin\phi_A + C_{M_P} \frac{pd}{2V_A} \sin\phi_A + C_{m_q} \frac{qd}{2V_A} - C_{m_{\delta_A}} \delta_A$$

$$C_{NG} = C_n \cos\phi_A - C_m \sin\phi_A + C_{n_p} \frac{pd}{2V_A} \cos\phi_A + C_{m_q} \frac{rd}{2V_A} + C_{M_{\delta_B}} \delta_B$$

Then, the moments about the center of gravity can be written as:

$$\begin{bmatrix} M_{LG} \\ M_{MG} \\ M_{NG} \end{bmatrix} = \begin{bmatrix} C_{L_G} + C_{Y_G} \Delta z - C_{Z_G} \Delta Y \\ C_{M_G} + C_{Z_G} \Delta X \\ C_{N_G} - C_{Y_G} \Delta X \end{bmatrix} \qquad (Q\ A\ d)$$

where ΔX, ΔY, ΔZ are the nondimensional lengths (ft/D) from the origin of the geometric axes to the origin of the principal axes (see Figure 8).

MOD8 Parameters

Parameter	Units	Location in Y Array
Input		
$\Delta X, \Delta Y, \Delta Z$		3006, 3007, 3008
X_R	ft	803
Y_R	ft	804
Z_R	ft	805
R_E	ft	3000
$\dot{X}_R$	ft/sec	800
$\dot{Y}_R$	ft/sec	801
$\dot{Z}_R$	ft/sec	802
ω_E	rad/sec	3001
ψ_R	deg	3015
ϕ_G	deg	3002
θ_G	deg	3003
d	ft	3004
A	ft^2	3005
δ	deg	604
δ_A	deg	386
δ_B	deg	387
ℓ_{LP}		2007-2035
ℓ_{RL}		2000-2008
p,q,r		806, 807, 808
Output		
M		577
Q	lb/ft^2	576
α	deg	572
β	deg	573
$\overline{\alpha}$	deg	599
ϕ_A	deg	574
F_{X_R}	lb	550
F_{Y_R}	lb	551
F_{Z_R}	lb	552

Parameter	Units	Location in Y Array
M_{LP}	ft-lb	547
M_{MP}	ft-lb	548
M_{NP}	ft-lb	
ℓ_{GP}	ft-lb	563-571
V_A	fps	575

MOD8 TABLES

Table Array No.	Table	Units
3	$V_W(h)$	fps
4	$A_W(h)$	deg
12	$C_x(M, \bar{\alpha})$	
13	$C_y(M, \bar{\alpha}, \phi_A)$	
14	$C_z(M, \bar{\alpha}, \phi_A)$	
15	$C_{y_p}(M, \bar{\alpha})$	
16	$C_l(M, \bar{\alpha}, \phi_A)$	
17	$C_{l_p}(M, \bar{\alpha})$	
18	$C_{l_\delta}(M, \bar{\alpha})$	
19	$C_m(M, \bar{\alpha}, \phi_A)$	
20	$C_M(M, \bar{\alpha}, \phi_A)$	
21	$C_{m_q}(M, \bar{\alpha})$	
22	$C_{Mp}(M, \bar{\alpha})$	
23	THRUST (t)	lb
24	$m_s(t)$	slug
25	$I_{xx}(t)$	slug/ft^2
26	$I_{yy}(t)$	slug/ft^2
27	$I_{zz}(t)$	slug/ft^2
28	$\Delta X(t)$	
29	$\Delta X(t)$	
30	$\Delta Z(t)$	

```
      SUBROUTINE MOD8
C
C     A GENERAL PURPOSE 6DOF FORCE AND MOMENT MODULE FOR A
C     MISSILE WITH MOMENT CONTROL IN TWO PLANES
C                                                                       S1101030
C        3/10/78
C                                                                       S1101040
C     TABLES.  KTAB(3)=VW(H) . WIND SPEED (FPS)                         S1101050
C              KTAB(4)=AW(H) . WIND AZIMUTH (DEG)                       S1101060
C              KTAB(12)=CX(M.ALPHA BAR)                                 S1101070
C              KTAB(13)=CY(M.ALPHA BAR.PHIA)                            S1101080
C              KTAB(14)=CZ(M.ALPHA BAR.PHIA)                            S1101090
C              KTAB(15)=CYP(M.ALPHA BAR)                                S1101100
C              KTAB(16)=CL(M.ALPHA BAR.PHIA)                            S1101110
C              KTAB(17)=CLP(M.ALPHA BAR)                                S1101120
C              KTAB(18)=CLD(M.ALPHA BAR)                                S1101130
C              KTAB(19)=CM(M.ALPHA BAR.PHIA)                            S1101140
C              KTAB(20)=CN(M.ALPHA BAR.PHIA)                            S1101150
C              KTAB(21)=CMQ(M.ALPHA BAR)                                S1101160
C              KTAB(22)=CNP(M.ALPHA BAR)                                S1101170
C              KTAB(23)=THRUST(T).  (LB)                                S1101180
C              KTAB(24)=MS(T).  TOTAL MASS.(SLUG)                       S1101190
C              KTAB(25)=IXX(T).  (SLUG-FT**2)                           S1101200
C              KTAB(26)=IYY(T).  (SLUG-FT**2)                           S1101210
C              KTAB(27)=IZZ(T).  (SLUG-FT**2)                           S1101220
C              KTAB(28)=DXG(T)
C              KTAB(29)=DYG(T)
C              KTAB(30)=DZG(T)
C                                                                       S1101260
C     D=REFERENCE LT. (FT)                                              S1101270
C     A=REFERENCE AREA (FT**2)                                          S1101280
C     DXG *LOCATION OF GEOM. AXES W/R TO AERODYN. DATA AXES
C     DYG *
C     DZG *(NONDIMENSIONAL.  FT/D)
C     PHG YAW ANGLE (DEG) BETWN GEOM. AXES AND PRIN. AXES               S1101320
C     THG PITCH ......................................                  S1101330
C
C
      COMMON Y(4940)
      COMMON/TAB/Z(50)                                                  S1101350
      EQUIVALENCE(Y(578),GRAV).(Y(577),VMACH)                           S1101360
      EQUIVALENCE(Y(800).XRD).(Y(801).YRD).(Y(802).ZRD)                 S1101370
      EQUIVALENCE(Y(3015),PSTR),(Y(3014).TAUR)                          S1101380
      EQUIVALENCE(Y(3001),WE).(Y(3000),RE).(Y(2027).LLP(1))             S1101390
      EQUIVALENCE(Y(2000).LRL(1)),(Y(3013).H).(Y(563).LGP(1))           S1101400
      EQUIVALENCE(Y(3019),THETAG).(Y(3020).PSIG).(Y(3021).PHIG)
      EQUIVALENCE(Y(803).XR),(Y(804),YR),(Y(805).ZR)                    S1101420
      EQUIVALENCE(Y(806).P).(Y(807).Q).(Y(808),R)                       S1101430
      EQUIVALENCE(Y(3004).D).(Y(3005).A)                                S1101440
      EQUIVALENCE(Y(3006).DXG),(Y(3007).DYG).(Y(3008).DZG)              S1101450
      EQUIVALENCE(Y(547).MLP),(Y(548).MMP).(Y(549).MNP)                 S1101460
      EQUIVALENCE(Y(550).FXR).(Y(551).FYR).(Y(552).FZR)                 S1101470
      EQUIVALENCE(Y(512).VXLE).(Y(513).VYLE).(Y(514).VZLE)              S1101480
      EQUIVALENCE(Y(500).XED),(Y(501).YED).(Y(502).ZED)                 S1101490
      EQUIVALENCE(Y(2057).LPL(1)),(Y(2039).LLR(1)).(Y(2048).LPG(1))     S1101500
      EQUIVALENCE(Y(572).ALPHA).(Y(573).BETA).(Y(574).PHIA)             S1101510
      EQUIVALENCE(Y(575).VA).(Y(576).QP),(Y(520).VAXL)                  S1101520
      EQUIVALENCE(Y(521).VAYL).(Y(522).VAZL)                            S1101530
      EQUIVALENCE(Y(3451).K(1))                                         S1101540
      EQUIVALENCE(Y(2999).T)                                            S1101550
      EQUIVALENCE(Y(544).FXG),(Y(545).FYG).(Y(546).FZG)                 S1101560
      EQUIVALENCE(Y(590).THRUST).(Y(599).ALPH)                          S1101570
```

```
      EQUIVALENCE(Y(529),CX),(Y(532),CY),(Y(533),CZ)                    S1101580
      EQUIVALENCE(Y(534),CLP),(Y(536),CM),(Y(537),CMQ)                  S1101590
      EQUIVALENCE(Y(535),CLD),(Y(539),CN)                               S1101600
      EQUIVALENCE(Y(614),CYP),(Y(615),CNP),(Y(616),CL)                  S1101610
      EQUIVALENCE(Y(518),VW),(Y(519),AW)                                S1101620
      EQUIVALENCE(Y(2037),THETAL),(Y(3012),MS)                          S1101630
      EQUIVALENCE(Y(526),VAXG),(Y(527),VAYG),(Y(528),VAZG)              S1101640
      EQUIVALENCE(Y(2009),LRP),(Y(556),PHIAD),(Y(604),DAF)              S1101650
      EQUIVALENCE(Y(605),PD2V),(Y(606),QD2V),(Y(607),RD2V)              S1101660
      EQUIVALENCE(Y(386),DA),(Y(387),DB)                                S1101670
      EQUIVALENCE(Y(608),CMDA),(Y(609),CNDB)                            S1101680
      EQUIVALENCE(Y(611),MLG),(Y(612),MMG),(Y(613),MNG)                 S1101690
      EQUIVALENCE(Y(3009),IXX),(Y(3010),IYY),(Y(3011),IZZ)              S1101700
      EQUIVALENCE(Y(610),VE)                                            S1101710
      DIMENSION K(49)                                                   S1101720
      DIMENSION B(9)                                                    S1101730
      DIMENSION U(3)                                                    S1101740
      REAL LLR(9),LPG(9),LPL(9)                                         S1101750
      REAL LRL(9),LGP(9),MLP,MMP,MNP,LLP(9)                             S1101760
      REAL LRP(9),MS,IXX,IYY,IZZ                                        S1101770
      REAL MLG,MMG,MNG                                                  S1101780
      RAD=3.141592653589/180.                                           S1101790
C
C     VELOCITIES
C
10    CALL MATVEC(LRL(1),Y(800),Y(512),0)                               S1101800
      H=SQRT(XR**2+YR**2+ZR**2)-RE                                      S1101810
      VXLE=Y(512)                                                       S1101820
      VYLE=Y(513)+WE*(RE+H)*COS(RAD*PSIR)                               S1101830
      VZLE=Y(514)                                                       S1101840
      VE=SQRT(VXLE**2+VYLE**2+VZLE**2)                                  S1101850
      CALL ITAB(3,1,H,VW)                                               S1101860
      CALL ITAB(4,1,H,AW)                                               S1101870
      AW=AW*RAD                                                         S1101880
   60 VWXLE=-VW*COS(AW)                                                 S1101890
      VWYLE=VW*SIN(AW)                                                  S1101900
      VWZLE=0.0                                                         S1101910
   70 VAXL=VXLE-VWXLE                                                   S1101920
      VAYL=VYLE-VWYLE                                                   S1101930
      VAZL=VZLE-VWZLE                                                   S1101940
80    CALL MATVEC(Y(2027),Y(520),Y(523),0)                              S1101950
C
C     PRINCIPAL AXIS MISALIGNMENT
C
      CALL SENCOS(THETAG,STG,CTG,0)
      CALL SENCOS(PSIG,SPG,CPG,0)
      CALL SENCOS(PHIG,SPHG,CPHG,0)
      LGP(1)=CPG*CTG
      LGP(2)=CPHG*STG-SPHG*SPG*CTG
      LGP(3)=-SPHG*STG-CPHG*SPG*CTG
      LGP(4)=-CPG*STG
      LGP(5)=CPHG*CTG+SPHG*SPG*STG
      LGP(6)=-SPG*CTG+CPHG*SPG*STG
      LGP(7)=SPG
      LGP(8)=SPHG*CPG
      LGP(9)=CPHG*CPG
      CALL MATINV(LGP,B,LPG)                                            S1102070
100   CALL MATVEC(LPG,Y(523),Y(526),0)                                  S1102080
C
C     ATMOSPHERIC/FLOW PROPERTIES
C
      VA=SQRT(Y(526)**2+Y(527)**2+Y(528)**2)                            S1102090
      IF(H.GE.500000.) GO TO 115                                        S1102100
110   CALL ARDCFT(H,PP,TT,DD,VS,G)                                      S1102110
      VMACH=VA/(VS*1116.4)                                              S1102120
      RHO=DD*0.0023769                                                  S1102130
```

```
      QP=0.5*RHO*VA**2                                                  S1102140
      GO TO 118                                                         S1102150
115   VMACH=0.0                                                         S1102160
      RHO=0.0                                                           S1102170
      QP=0.0                                                            S1102180
118   GRAV=(32.174*RE**2)/((SQRT(XR**2+YR**2+ZR**2))**2)                S1102190
C
C     ANGULAR RELATIONSHIP BETWEEN MISSILE AND VELOCITY VECTOR
C
  120 CALL ARKTAN(Y(528),Y(526),ALPHA,0)                                S1102200
      CALL ARKTAN(Y(527),Y(526),BETA,0)                                 S1102210
      CALL ARKTAN(Y(527),Y(528),PHIA,0)                                 S1102220
      CALL ARKTAN((SQRT(Y(527)**2+Y(528)**2)),Y(526),ALPB,0)            S1102230
C                                                                       S1102240
C     FORCE AND MOMENT GENERATION  ************************************
C                                                                       S1102260
C     TABULATED AERODYNAMIC COEFFICIENTS
C
      U(1)=VMACH                                                        S1102270
      U(2)=ALPB                                                         S1102280
      CALL TTAB(12,2,U,CX)                                              S1102290
      CALL TTAB(15,2,U,CYP)                                             S1102300
      CALL TTAB(17,2,U,CLP)                                             S1102310
      CALL TTAB(18,2,U,CLD)                                             S1102320
      CALL TTAB(21,2,U,CMQ)                                             S1102330
      CALL TTAB(22,2,U,CNP)                                             S1102340
      IF(PHIA) 140,150,150                                              S1102350
140   PHIA=PHIA+360.                                                    S1102360
150   CONTINUE                                                          S1102370
      U(3)=AMOD(PHIA,90.)                                               S1102380
                                                                        S1102390
      CALL TTAB(13,3,U,CY)                                              S1102400
      CALL TTAB(14,3,U,CZ)                                              S1102410
      CALL TTAB(16,3,U,CL)                                              S1102420
      CALL TTAB(19,3,U,CM)                                              S1102430
      CALL TTAB(20,3,U,CN)                                              S1102440
C
C     ANGULAR RATES OF BODY W/R TO FLOW
C
      CALL SENCOS(PHIA,SPH,CPH,0)                                       S1102450
      CALL MATVEC(LRP,Y(812),Y(553),0)                                  S1102460
      CALL MATVEC(LPG,Y(553),Y(553),0)                                  S1102470
      CALL MATVEC(LPG,Y(806),Y(820),0)                                  S1102480
      IF(VAZG**2+VAYG**2) 160,160,170                                   S1102490
160   PHIAD=Y(820)                                                      S1102500
      GO TO 175                                                         S1102510
170   CONTINUE                                                          S1102520
      PHIAD=(VAZG*(Y(554)-Y(822)*VAXG)-VAYG*(Y(555)+Y(821)*VAXG))       S1102530
     +/(VAZG**2+VAYG**2)+Y(820)                                         S1102540
175   CONTINUE                                                          S1102550
      PD2V=PHIAD*D/(2.0*VA)                                             S1102560
      QD2V=Y(821)*D/(2.0*VA)                                            S1102570
      RD2V=Y(822)*D/(2.0*VA)                                            S1102580
C
C     AERO COEFFICIENTS
C
      CXG=CX                                                            S1102590
      CYG=CY*CPH+CZ*SPH+CYP*PD2V*CPH                                    S1102600
      CZG=CZ*CPH-CY*SPH-CYP*PD2V*SPH                                    S1102610
      CLG=CL+CLP*PD2V
      CMG=CM*CPH+CN*SPH+CNP*PD2V*SPH+CMQ*QD2V-CMDA*DA                   S1102630
      CNG=CN*CPH-CM*SPH+CNP*PD2V*CPH+CMQ*RD2V+CNDR*DR                   S1102640
C
C     THRUST AND MASS PROPERTIES
C
      U(1)=T                                                            S1102650
```

```
      CALL ITAB(23,1,U,THRUST)                                          S1102660
      CALL ITAB(24,1,U,MS)                                              S1102670
      CALL ITAB(25,1,U,IXX)                                             S1102680
      CALL ITAB(26,1,U,IYY)                                             S1102690
      CALL ITAB(27,1,U,IZZ)                                             S1102700
C
C     FORCES AND MOMENTS
C
      FXG=CXG*QP*A+THRUST                                               S1102710
      FYG=CYG*QP*A                                                      S1102720
      FZG=CZG*QP*A                                                      S1102730
      CALL ITAB(28,1,U,DXG)                                             S1102740
      CALL ITAB(29,1,U,DYG)                                             S1102750
      CALL ITAB(30,1,U,DZG)                                             S1102760
      MLG=(CLG+CYG*DZG-CZG*DYG)*QP*A*D                                  S1102770
      MMG=(CMG+CZG*DXG)*QP*A*D                                          S1102780
      MNG=(CNG-CYG*DXG)*QP*A*D                                          S1102790
      CALL MATVEC(LGP,Y(611),Y(547),0)                                  S1102800
  200 CONTINUE                                                          S1102810
      CALL MATVEC(LGP,Y(544),Y(550),0)                                  S1102820
      CALL MATVEC(LPL,Y(550),Y(550),0)                                  S1102830
      CALL MATVEC(LLR,Y(550),Y(550),0)                                  S1102840
                                                                        S1102850
      RETURN                                                            S1102860
      END
```

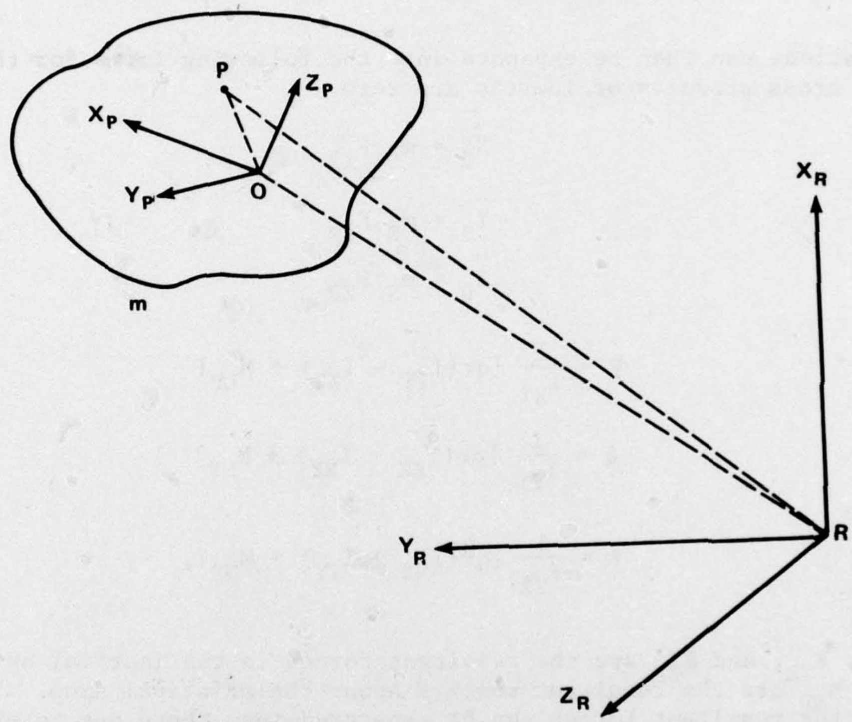
THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

APPENDIX N

MOD9; EQUATIONS OF MOTION MODULE

The purpose of this module is to express the equations of motion for a 6DOF simulation of a fairly general body. The force equations are written for integration in the inertial frame and the moment equations for integration in the principal axes.

For a mass, m, with the following characteristics,



where X_P, Y_P, Z_P are the principal axes of the mass, and X_R, Y_R, Z_R are the inertial axes, the general acceleration of point, p is,

$$(\vec{a})_R = (\vec{a}_o)_R + \vec{\omega} \times (\vec{\omega} \times \vec{OP}) + \dot{\vec{\omega}} \times \vec{OP} + (\vec{a}_P)_o + 2\vec{\omega} \times (\vec{V}_P)_o$$

In this equation, $(\vec{a}_P)_R$ is the acceleration of P with respect to R,

$(\vec{a}_o)_R$ is the acceleration of o with respect to R,

$(\vec{a}_P)_o$ is the acceleration of P with respect to o, and

$(\vec{V}_P)_o$ is the velocity of P with respect to o.

If the assumption is made that m is a rigid body, then

$$(\vec{a}_P) = 0$$

$$(\vec{V}_P)_o = 0$$

and if 0 is the center of gravity of m, then the following force and moment equations can be written.

$$F = \int_m (\vec{a})_r \, dm = m(\vec{a}_o)_R$$

$$\vec{M} = \int_m [\vec{OP} \times (\vec{a})_R] dm = \int_m \vec{OP} \times [\dot{\vec{\omega}} \times \vec{OP} + \vec{\omega} \times (\vec{\omega} \times \vec{OP})] dm$$

These equations can then be expanded into the following forms for the case where the cross products of inertia are zero.

$$\ddot{X}_R = m_S \, F_{XR}$$

$$\ddot{Y}_R = m_S \, F_{YR}$$

$$\ddot{Z}_R = m_S \, F_{ZR}$$

$$\dot{P} = \frac{1}{I_{XX}} [qr(I_{YY} - I_{ZZ}) + M_{LP}]$$

$$\dot{q} = \frac{1}{I_{YY}} [pr(I_{ZZ} - I_{XX}) + M_{MP}]$$

$$\dot{r} = \frac{1}{I_{ZZ}} [qP(I_{XX} - I_{YY}) + M_{NP}],$$

where F_{XR}, F_{YR}, and F_{ZR} are the resultant forces in the inertial system, and M_{LP}, M_{MP}, M_{NP} are the resultant moments about the principal axes. In the force equations the resultant forces can be separated into those due to gravity and those due to all other forces. The force equations then become,

$$\ddot{X}_R = \frac{F_{XR}}{m_S} - g \sin\psi_R$$

$$\ddot{Y}_R = \frac{F_{YR}}{m_S} + g \cos\psi_R \sin\tau_R$$

$$\ddot{Z}_R = \frac{F_{ZR}}{m_S} - g \cos\psi_R \cos\tau_R,$$

where F_{XR}, F_{YR}, F_{ZR} consist of all external forces except those caused by gravity.

MOD9 Parameters

	Parameter	Units	Location in Y Array
Input	ψ_R	degree	3015
	τ_R	degree	3014
	F_{XR}	pound	550
	F_{YR}	pound	551
	F_{ZR}	pound	552
	m_S	slug	3012
	GRAV	feet/second^2	578
	P	rad/sec	806
	q	rad/sec	807
	r	rad/sec	808
	I_{XX}	slug-ft^2	3009
	I_{YY}	slug-ft^2	3010
	I_{ZZ}	slug-ft^2	3011
	M_{LP}	feet/pound	547
	M_{MP}	feet/pound	548
	M_{NP}	feet/pound	549
Output	$\ddot{X}_R$	feet/second^2	812
	$\ddot{Y}_R$	feet/second^2	813
	$\ddot{Z}_R$	feet/second^2	814
	$\dot{P}$	rad/sec^2	815
	$\dot{q}$	rad/sec^2	816
	$\dot{r}$	rad/sec^2	817

```
      SUBROUTINE MOD9                                                   SIX06450
C                                                                       SIX06460
C        3/10/78
C                                                                       SIX06480
C     EQUATIONS OF MOTION                                               SIX06490
C                                                                       SIX06500
C     MOD PACKAGE SIXDG. A GENERAL PURPOSE 6DOF GUIDED OR               SIX06510
C     UNGUIDED TRAJECTORY PROGRAM                                       SIX06520
C                                                                       SIX06530
C     THE MASS,MS AND MOMENTS OF INERTIA IXX,IYY,IZZ ARE REQUIRED       SIX06540
C                                                                       SIX06550
C     NOTE"  DERIVATIVES XRDD,YRDD,ZRDD.XRD,YRD,ZRD,PD,QD.RD,P,Q,R      SIX06560
C                                                                       SIX06570
      COMMON Y(4940)                                                    SIX06580
      EQUIVALENCE(Y(547),MLP),(Y(548),MMP),(Y(549),MNP)                 SIX06590
      EQUIVALENCE(Y(550),FXR),(Y(551),FYR),(Y(552),FZR)                 SIX06600
      EQUIVALENCE(Y(812),XRDD),(Y(813),YRDD),(Y(814),ZRDD)              SIX06610
      EQUIVALENCE(Y(815),PD),(Y(816),QD),(Y(817),RD)                    SIX06620
      EQUIVALENCE(Y(3009),IXX),(Y(3010),IYY),(Y(3011),IZZ)              SIX06630
      EQUIVALENCE(Y(806),P),(Y(807),Q),(Y(808),R)                       SIX06640
      EQUIVALENCE(Y(3012),MS)                                           SIX06650
      EQUIVALENCE(Y(3015),PSIR),(Y(3014),TAUR),(Y(578),GRAV)            SIX06660
      EQUIVALENCE(Y(2999),T)                                            SIX06670
      REAL MS,MLP,MMP,MNP,IXX,IYY,IZZ                                   SIX06680
C
C     LINEAR EQUATIONS OF MOTION
C
      CALL SENCOS(PSIR,SPS,CPS,0)                                       SIX06690
      CALL SENCOS(TAUR,STA,CTA,0)                                       SIX06700
      XRDD=FXR/MS-GRAV*SPS                                              SIX06710
      YRDD=FYR/MS+GRAV*CPS*STA                                          SIX06720
      ZRDD=FZR/MS-GRAV*CPS*CTA                                          SIX06730
C
C     ANGULAR EQUATIONS OF MOTION
C
30    PD=(Q*R*(IYY-IZZ)+MLP)/IXX                                        SIX06740
      QD=(P*R*(IZZ-IXX)+MMP)/IYY                                        SIX06750
      RD=(Q*P*(IXX-IYY)+MNP)/IZZ                                        SIX06760
40    CONTINUE                                                          SIX06770
      RETURN                                                            SIX06780
      END                                                               SIX06790
```

APPENDIX O

MOD14; 3DOF FORCE AND EQUATIONS OF MOTION MODULE

The purpose of this module is to calculate the forces acting on the vehicle flying along a particle (3DOF) trajectory and to set up the equations of motion.

If winds are desired, they may be input in tabular form. The wind velocity, V_w, and the heading angle, A_w (measured clockwise from the north) can be tabulated as a function of altitude, h in TABLE ARRAYS NO. 3 and 4 respectively. In that case the velocity of the vehicle with respect to the air is,

$$\begin{bmatrix} V_{A_{XL}} \\ V_{A_{YL}} \\ V_{A_{ZL}} \end{bmatrix} = \begin{bmatrix} V_{X_{LE}} \\ V_{Y_{LE}} \\ V_{Z_{LE}} \end{bmatrix} - \begin{bmatrix} -V_w \cos A_w \\ V_w \sin A_w \\ 0 \end{bmatrix}$$

or

$$V_A = \sqrt{{V_{A_{XL}}}^2 + {V_{A_{YL}}}^2 + {V_{A_{ZL}}}^2}$$

The atmospheric properties such as the speed of sound and density are gotten from the 1969 Standard Atmospheric Properties Tables.

The forces along the body principal axes are then expressed as

$$F_{XP} = \text{THRUST} - \left(\frac{1}{2}\rho {V_A}^2\right)\left(A_{REF}\right)\left(C_D\right)$$

$$F_{YP} = 0$$

$$F_{ZP} = 0$$

C_D is the drag coefficient input in tabular form as a function of Mach number in TABLE ARRAY No. 5.

Other associated necessary quantities are the mass of the vehicle and the acceleration due to gravity. These are,

$$m_S = m_I + \dot{m}(t - t_i)$$

where

m_I = initial mass

$\dot{m}$ = mass depletion rate

t_i = initial time

m_S = system mass

t = current time

and

$$g = \frac{g_o R_E^{\,2}}{\left(\sqrt{X_R^{\,2} + Y_R^{\,2} + Z_R^{\,2}}\right)^2}$$

The equations of motion can then be written as,

$$\begin{bmatrix} \ddot{X}_R \\ \ddot{Y}_R \\ \ddot{Z}_R \end{bmatrix} = [\ell_{LR}][\ell_{PL}] \begin{bmatrix} F_{XP} \\ F_{YP} \\ F_{ZP} \end{bmatrix} + g \begin{bmatrix} \sin\psi_R \\ \cos\psi_R \sin\tau_R \\ \cos\psi_R \cos\tau_R \end{bmatrix}$$

MOD14 Parameters

	Parameter	Units	Location in Y Array
Input			
	H	ft	3013
	$V_{X_{LE}}$	fps	512
	$V_{Y_{LE}}$	fps	513
	$V_{Z_{LE}}$	fps	514
	X_R	ft	803
	Y_R	ft	804
	Z_R	ft	805
	M_I	slug	658
	$\dot{m}$	slug/sec	589
	t_i	sec	2863
	t	sec	2999
	THRUST	lb	590
Output			
	$V_{A_{XL}}$	fps	520
	$V_{A_{YL}}$	fps	521
	$V_{A_{ZL}}$	fps	522
	V_A	fps	575
	M	fps	577
	C_D	fps	529
	F_{XP}	lb	601
	F_{YP}	lb	602
	F_{ZP}	lb	603
	F_{XR}	lb	550
	F_{YR}	lb	551
	F_{ZR}	lb	552
	$\ddot{X}_R$	ft/sec^2	812
	$\ddot{Y}_R$	ft/sec^2	813
	$\ddot{Z}_R$	ft/sec^2	814

```
      SUBROUTINE MOD14
C
C          FORCE MODULE FOR A 3DOF PARTICAL TRAJ.
C
C        3/10/78
C
C     TABLES,  KTAB(3)=VW(H) , WIND SPEED (FPS)                          TR301360
C              KTAB(4)=AW(H) , WIND AZIMUTH (DEG)                        TR301370
C              KTAB(5)=CD(M)   ,DRAG COEFF.
C              KTAB(23)=THRUST(T)    (LB)
C
C
C     INPUT PARAMETERS
C
C     MI=Y(658)    ,   INITIAL MASS (SLUG)
C     MDOT=Y(589),   RATE OF CHANGE OF MASS (SLUGS/SEC)
C     TI=Y(2862)   .   INITIAL TIME (SEC)
C     THRUST=Y(590)    ,   CONSTANT THRUST (LB)
C     A=Y(3005)    ,   REF. AREA (FT**2)
C
      COMMON Y(4940)
      EQUIVALENCE(Y(3013),H),(Y(2999),T)
      EQUIVALENCE(Y(512),VXLE),(Y(513),VYLE),(Y(514),VZLE)
      EQUIVALENCE(Y(2057),LPL(1)),(Y(2039),LLR(1))
      EQUIVALENCE(Y(803),XR),(Y(804),YR),(Y(805),ZR)
      EQUIVALENCE(Y(578),GRAV),(Y(577),VMACH)                            TR301450
      EQUIVALENCE(Y(812),XRDD),(Y(813),YRDD),(Y(814),ZRDD)               TR301480
      EQUIVALENCE(Y(3012),MS)                                            TR301520
      EQUIVALENCE(Y(550),FXR),(Y(551),FYR),(Y(552),FZR)                  TR301530
      EQUIVALENCE(Y(3005),A),(Y(3000),RE)
      EQUIVALENCE(Y(349),PSIR),(Y(348),TAUR)
      EQUIVALENCE(Y(575),VA),(Y(576),QP),(Y(520),VAXL)                   TR301580
      EQUIVALENCE(Y(521),VAYL),(Y(522),VAZL)                             TR301590
      EQUIVALENCE(Y(601),FXP),(Y(602),FYP),(Y(603),FZP)                  TR301620
      EQUIVALENCE(Y(589),MDOT),(Y(590),THRUST)                           TR301630
      EQUIVALENCE(Y(529),CD)                                             TR301640
      EQUIVALENCE(Y(658),MI),(Y(2862),TI)                                TR301670
      DIMENSION U(2)
      REAL LLR(9),LPL(9),MS,MI,MDOT
C
      RAD=3.141592653589/180.                                            TR301820
C
C          WINDS
C
      CALL ITAB(3,1,H,VW)
      CALL ITAB(4,1,H,AW)
      AW=AW*RAD                                                          TR301830
   60 VWXLE=-VW*COS(AW)                                                  TR301840
      VWYLE=VW*SIN(AW)                                                   TR301850
      VWZLE=0.0                                                          TR301860
C
C          VELOCITY W/R TO AIR
C
   70 VAXL=VXLE-VWXLE                                                    TR301870
      VAYL=VYLE-VWYLE                                                    TR301880
      VAZL=VZLE-VWZLE                                                    TR301890
      VA=SQRT(Y(520)**2+Y(521)**2+Y(522)**2)                             TR301900
C
C          ENVIRONMENTAL PROPERTIES
C
      IF(H.GE.500000.) GO TO 115                                         TR301910
  110 CALL ARDCFT(H,PP,TT,DD,VS,G)                                       TR301920
```

```
      VMACH=VA/(VS*1116.4)                                              TR301930
      RHO=DD*0.0023769
      QP=0.5*RHO*VA**2                                                  TR301970
      GO TO 118                                                         TR301980
115   VMACH=0.0                                                         TR301990
      RHO=0.0                                                           TR302000
      QP=0.0                                                            TR302010
118   GRAV=(32.174*RE**2)/((SQRT(XR**2+YR**2+ZR**2))**2)                TR302020
C                                                                       TR302030
C          MASS
C
      MS=MI+MDOT*(T-TI)                                                 TR302080
C
C          FORCES
C
      U(1)=VMACH                                                        TR302090
      CALL ITAB(5,1,U,CD)
      FXP=THRUST-QP*A*CD                                                TR302130
      FYP=0.0                                                           TR302140
      FZP=0.0                                                           TR302150
      CALL MATVEC(LPL,Y(601),Y(550),0)                                  TR302270
      CALL MATVEC(LLR,Y(550),Y(550),0)                                  TR302280
C
C          EQUATIONS OF MOTION
C
      XRDD=FXR/MS-GRAV*    (SIN(PSIR))                                  TR302290
      YRDD=FYR/MS+GRAV*    (COS(PSIR)*SIN(TAUR))                        TR302300
      ZRDD=FZR/MS-GRAV*    (COS(PSIR)*COS(TAUR))                        TR302310
      RETURN
      END
```

APPENDIX P

PROCESS, 3DOF AND 6DOF

PROCESS PARAMETERS

Input	Parameter	Units	Location in Y Array
	RE	FT	3000
	WE	rad/sec	3001
	TAUR	deg.	3014
	PSIR	deg.	3015
	TI	sec.	2862
	T	sec.	2999
	VXLE	fps	512
	VYLE	fps	513
	VZLE	fps	514
Output			
	GX	g's	2205
	GY	g's	2206
	GZ	g's	2207
	GAMAH	deg.	2208
	GAMAE	deg.	2209

```
      SUBROUTINE PROCESS                                                SIX06820
C                                                                       SIX06830
C        3/13/78
C                                                                       SIX06850
C     MOD PACKAGE SIXDG. A GENERAL PURPOSE 6DOF GUIDED OR               SIX06860
C     UNGUIDED TRAJECTORY PROGRAM                                       SIX06870
C                                                                       SIX06880
      COMMON Y(4940)                                                    SIX06890
      EQUIVALENCE(Y(2863),STOP),(Y(3012),MS),(Y(658),MI)                SIX06900
      EQUIVALENCE(Y(2213),OT),(Y(2214),OP),(Y(2215),RS)                 SIX06910
      EQUIVALENCE(Y(3000),RE),(Y(2862),TI),(Y(3001),WE)                 SIX06920
      EQUIVALENCE(Y(504),OLAL),(Y(505),OLB),(Y(506),OTIM),(Y(2999),T)
      EQUIVALENCE(Y(3014),TAUR),(Y(3015),PSIR)                          SIX06940
      EQUIVALENCE(Y(818),RTEI),(Y(819),RPEI)                            SIX06950
      EQUIVALENCE(Y(803),XRM),(Y(804),YRM),(Y(805),ZRM)                 SIX06960
      EQUIVALENCE(Y(2200),RTE),(Y(2201),RPE)                            SIX06970
      EQUIVALENCE(Y(2205),GX),(Y(2206),GY),(Y(2207),GZ)                 SIX06980
      EQUIVALENCE(Y(512),VXLE),(Y(513),VYLE),(Y(514),VZLE)              SIX06990
      EQUIVALENCE(Y(2208),GAMAH),(Y(2209),GAMAE)                        SIX07000
      EQUIVALENCE (Y(2204),GT)
      EQUIVALENCE(Y(651),CXA),(Y(652),CYA),(Y(653),CZA)
      EQUIVALENCE(Y(526),VXG),(Y(527),VYG),(Y(528),VZG)
      EQUIVALENCE(Y(648),CXG),(Y(649),CYG),(Y(650),CZG)
      EQUIVALENCE(Y(660),LGA(1)),(Y(3005),A),(Y(578),GRAV)
      EQUIVALENCE(Y(3012),MS),(Y(654),BALC)
      REAL MI,MS,LGA(9)
C                                                                       SIX07020
C     EQUATORIAL RANGES                                                 SIX07030
C
      RAD=3.141592653589/180.                                           SIX07040
      RTE=(RE*(TAUR*RAD-WE*(T-TI)))                                     SIX07050
      RPE=(RE*PSIR*RAD)                                                 SIX07060
C                                                                       SIX07070
C     BODY ACCELERATIONS IN THE PRINCIPAL AXES, (G,S)                   SIX07080
C
      CALL MATVEC(Y(2009),Y(812),Y(2202),0)                             SIX07090
      GX=Y(2202)/32.174                                                 SIX07100
      GY=Y(2203)/32.174                                                 SIX07110
      GZ=Y(2204)/32.174                                                 SIX07120
      GT=SQRT(GY**2+GZ**2)
C                                                                       SIX07130
C     LOCAL FLIGHT PATH ANGLES,(DEG)                                    SIX07140
C
      CALL ARKTAN(-VYLE,VXLE,GAMAH,0)                                   SIX07150
      ZZZ=SQRT(VXLE**2+VYLE**2)                                         SIX07160
      CALL ARKTAN(VZLE,ZZZ,GAMAE,0)                                     SIX07170
C                                                                       SIX07180
C     TRUE DISTANCE TRAVELED W/R TO THE EARTHS SURFACE                  SIX07190
C
      TAUR=TAUR*RAD                                                     SIX07200
      PSIR=PSIR*RAD                                                     SIX07210
      TAUE=TAUR-WE*T                                                    SIX07220
      DX2=(RE*SIN(OP)-RE*SIN(PSIR))**2                                  SIX07230
      DY2=(-RE*COS(OP)*SIN(OT)+RE*COS(PSIR)*SIN(TAUE))**2               SIX07240
      DZ2=(RE*COS(OP)*COS(OT)-RE*COS(PSIR)*COS(TAUE))**2                SIX07250
      DC=SQRT(DX2+DY2+DZ2)                                              SIX07260
      THETA=2.0*ASIN(DC/(2.0*RE))                                       SIX07270
      DRS=RE*THETA                                                      SIX07280
      RS=RS+DRS                                                         SIX07290
C
      OT=TAUE                                                           SIX07300
      OP=PSIR                                                           SIX07310
```

```
      TAUR=TAUR/RAD                                                     SIX07320
      PSIR=PSIR/RAD                                                     SIX07330
      IF(STOP) 20,20,10                                                 SIX07340
   10 TI=T                                                              SIX07350
      MI=MS                                                             SIX07360
   20 CONTINUE                                                          SIX07370
      RETURN                                                            SIX07380
      END                                                               SIX07390
```

DISTRIBUTION

	Copies
Commander Naval Air Systems Command 1411 Jefferson Davis Highway Jefferson Plaza #1 Arlington, VA 20360 (ATTN: WILLIAM C. VOLZ)	5
Naval Sea Systems Command Washington, D. C. 20362 Attn: SEA-09G32 SEA-03B	2
Defense Documentation Center Cameron Station Alexandria, VA	12

NDW-NSWC(W)-5605/1 (Rev. 5-78)

TO AID IN UPDATING THE DISTRIBUTION LIST FOR NAVAL SURFACE WEAPONS CENTER, WHITE OAK TECHNICAL REPORTS PLEASE COMPLETE THE FORM BELOW:

TO ALL HOLDERS OF NSWC/WOL/TR 78-59
by John E. Holmes, Code K-81
DO NOT RETURN THIS FORM IF ALL INFORMATION IS CURRENT

A. FACILITY NAME AND ADDRESS (OLD) (Show Zip Code)

NEW ADDRESS (Show Zip Code)

B. ATTENTION LINE ADDRESSES:

C.

☐ REMOVE THIS FACILITY FROM THE DISTRIBUTION LIST FOR TECHNICAL REPORTS ON THIS SUBJECT.

D.
NUMBER OF COPIES DESIRED ______

GPO 880-764

DEPARTMENT OF THE NAVY
NAVAL SURFACE WEAPONS CENTER
WHITE OAK, SILVER SPRING, MD. 20910

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE NAVY
DOD 316

COMMANDER
NAVAL SURFACE WEAPONS CENTER
WHITE OAK, SILVER SPRING, MARYLAND 20910

ATTENTION: CODE K-81